



Documentazione di Sphinx

Release 7.4.0+/082f13f37 (Ita)

the Sphinx developers

(Traduzione: **Baldassarre Cesarano**)

08 lug 2024

1	Installazione di Sphinx	1
1.1	Pacchetto PyPI	2
1.2	Pacchetto Conda	2
1.3	Package manager specifico del sistema operativo	3
	Linux	3
	Debian/Ubuntu	3
	RHEL, CentOS	3
	Altre distribuzioni	3
	macOS	3
	Homebrew	4
	MacPorts	4
	Windows	4
	Chocolatey	4
1.4	Docker	4
1.5	Installazione dell'ultima versione di sviluppo	5
1.6	Installazione dai sorgenti	5
2	Per Iniziare	7
2.1	Impostazione dei sorgenti della documentazione	7
2.2	Definizione della struttura del documento	8
2.3	Aggiungere il contenuto	9
2.4	Esecuzione della build	9
2.5	Documentare gli oggetti	10
2.6	Configurazione di base	11
2.7	Autodoc	12
2.8	Intersphinx	12
2.9	Altri argomenti da trattare	12
3	La build del primo progetto	15
3.1	Per iniziare	16
	Configurazione del progetto e dell'ambiente di sviluppo	16
	Creazione del layout della documentazione	16
3.2	Primi passi per documentare il progetto utilizzando Sphinx	18

	La build della documentazione HTML	18
	Build della documentazione in altri formati	19
3.3	Altre personalizzazioni di Sphinx	20
	Abilitazione di un'estensione nativa	20
	Uso di un tema HTML di terze parti	20
3.4	Documentazione narrativa in Sphinx	21
	Strutturare la documentazione su più pagine	21
	Aggiungere i riferimenti incrociati	22
3.5	Descrizione del codice in Sphinx	24
	Python	24
	Documentare oggetti Python	24
	Riferimenti incrociati di oggetti Python	26
	Includere i doctest nella documentazione	27
	Altri linguaggi (C, C++, altri)	29
	Documentare e cross-referenziare gli oggetti	29
3.6	Generazione automatica della documentazione dal codice	30
	Riutilizzo di firme e docstring con autodoc	30
	Generazione di tutti i riferimenti delle API	31
3.7	Appendice: Distribuzione online di un progetto Sphinx	33
	Possibilità di distribuzione compatibili con Sphinx	33
	La filosofia "Docs as Code".	34
	Pubblicare i sorgenti della documentazione	35
	GitHub	35
	GitLab	35
	Pubblicare la documentazione HTML	36
	Read the Docs	36
	GitHub Pages	36
	GitLab Pages	38
3.8	Come proseguire	39
4	Uso di Sphinx	41
4.1	reStructuredText	41
	reStructuredText Primer	41
	Paragrafi	42
	Markup in linea	42
	Elenchi e blocchi evidenziati [quoted]	42
	Blocchi letterali	44
	Blocchi doctest	45
	Tabelle	45
	Hyperlink	46
	Sezioni	46
	Liste di campi	47
	Ruoli	47
	Markup Esplicito	48
	Direttive	48
	Immagini	51
	Note a piè di pagina	52
	Citazioni	52

	Sostituzioni	52
	Commenti	53
	Metadati HTML	53
	Codifica dei sorgenti	54
	Trabocchetti	54
Ruoli		54
	Sintassi dei riferimenti incrociati	55
	Evidenziazione del codice in linea	55
	Formule matematiche	56
	Altri markup semantici	56
	Sostituzioni	59
Direttive		60
	Indici e Sommari	60
	Markup a livello di paragrafo	64
	Mostrare esempi di codice	67
	Glossario	73
	Markup di meta-informazioni	75
	Markup che generano indici	75
	Inclusione di contenuto basato sui tag	78
	Tabelle	79
	Formule matematiche	80
	Visualizzazioni della produzione grammaticale	82
Liste di campi		83
	Metadati a livello di file	83
	Campi di metadati speciali	83
SPOSTATO: Domini		84
	SPOSTATO: Basic Markup	84
	SPOSTATO: Dominio Python	84
	SPOSTATO: Dominio C	84
	SPOSTATO: Dominio C++	84
	SPOSTATO: Dominio Standard	85
	SPOSTATO: Dominio JavaScript	85
	SPOSTATO: Dominio reStructuredText	85
	SPOSTATO: Dominio Math	85
	SPOSTATO: Altri domini	85
4.2	Markdown	85
	Configurazione	85
4.3	Sintassi dei riferimenti incrociati	86
	Riferimenti incrociati a qualsiasi cosa	87
	Riferimenti incrociati ad oggetti	87
	Riferimenti incrociati di posizioni arbitrarie	88
	Riferimenti incrociati a documenti	89
	Riferimento a file scaricabili	89
	Riferimenti incrociati a figure tramite il numero	90
	Riferimenti incrociati ad altri elementi	90
4.4	Configurazione	91
	Informazioni sul Progetto	92
	Configurazione generale	92

	Opzioni per l'internazionalizzazione	104
	Opzioni per le Formule	109
	Opzioni per l'output HTML	109
	Opzioni per l'output HTML Singolo	120
	Opzioni per l'output dell'help HTML	120
	Opzioni per l'output dell'Help di Apple	121
	Opzioni per l'output epub	123
	Opzioni per l'output LaTeX	127
	Opzioni per l'output text	133
	Opzioni per l'output della pagina del manuale	134
	Opzioni per l'output Texinfo	135
	Opzioni per l'output QtHelp	137
	Opzioni per il builder di linkcheck	137
	Opzioni per il builder XML	141
	Opzioni per il dominio C	141
	Opzioni per il dominio C++	142
	Opzioni per il dominio Python	142
	Esempi	142
	Opzioni per il dominio Javascript	143
	Esempio di file di configurazione	144
4.5	I Builder	152
	Dettagli del builder di serializzazione	163
4.6	Domini	165
	Markup di base	165
	Sintassi dei riferimenti incrociati	167
	Domini nativi	168
	Il Dominio Standard	168
	Il Dominio C	169
	Il Dominio C++	176
	Il Dominio JavaScript	189
	Il dominio della Matematica	192
	Il Dominio Python	192
	Il Dominio reStructuredText	203
	Altri domini	205
4.7	Estensioni	206
	Estensioni integrate	206
	sphinx.ext.autodoc -- Include la documentazione da docstring	206
	sphinx.ext.autosectionlabel -- Consente sezioni di riferimento utilizzandone il titolo	223
	sphinx.ext.autosummary -- Genera riepiloghi autodoc	224
	sphinx.ext.coverage -- Raccoglie statistiche sulla copertura dei documenti	229
	sphinx.ext.doctest -- Prova gli snippet nella documentazione	231
	sphinx.ext.duration -- Misura la durata dell'elaborazione di Sphinx	238
	sphinx.ext.extlinks -- Markup per abbreviare i link esterni	238
	sphinx.ext.githubpages -- Pubblica documenti HTML nelle pagine GitHub	239
	sphinx.ext.graphviz -- Aggiunge grafici Graphviz	240

	sphinx.ext.ifconfig -- Include il contenuto in base alla configura-	
	zione	243
	sphinx.ext.imgconverter -- Un convertitore di immagini di	
	referimento che utilizza Imagemagick	244
	sphinx.ext.inheritance_diagram -- Include diagrammi di eredi-	
	tarietà	245
	sphinx.ext.intersphinx -- Link alla documentazione di altri progetti	248
	sphinx.ext.linkcode -- Aggiunge link esterni al codice sorgente . .	253
	Supporto matematico per output HTML in Sphinx	254
	sphinx.ext.napoleon -- Supporto per docstring in stile NumPy e	
	Google	259
	sphinx.ext.todo -- Supporto per gli elementi todo	270
	sphinx.ext.viewcode -- Aggiunge i link al codice sorgente evidenziato	271
	Estensioni di terze parti	273
	Dove mettere le proprie estensioni?	273
4.8	Temi HTML	274
	I Builder	274
	Temi	274
	Usare un tema	274
	Temi nativi	275
	Temi di Terze Parti	281
4.9	Internazionalizzazione	281
	Dettagli sull'internazionalizzazione di Sphinx	282
	Tradurre con sphinx-intl	283
	Guida rapida	283
	Traduzione	285
	Aggiornare i file po con i nuovi file pot	285
	Utilizzo del servizio Transifex per team di traduzione	285
	Utilizzo del servizio Transifex per team di traduzione	288
	Contribuire alla traduzione dei reference di Sphinx	288
	Avanzamento e statistiche della traduzione	288
4.10	Supporto Web di Sphinx	288
	Quick Start del Supporto Web	289
	Building dei Dati della Documentazione	289
	Integrazione dei Documenti Sphinx Nella Propria Webapp	289
	Esecuzione delle Ricerche	292
	Commenti & Proposte	292
	Moderazione dei Commenti	293
	La Classe WebSupport	294
	Metodi	295
	I Search Adapter	298
	Metodi	299
	Gli Storage Backend	300
	Metodi	301
5	Estendere Sphinx	303
5.1	Tutorial	303
	Estendere la sintassi con ruoli e direttive	303

	Panoramica	303
	Impostazione del progetto	304
	Scrivere l'estensione	304
	Uso dell'estensione	307
	Ulteriori letture	308
	Estendere il processo di build	308
	Panoramica	308
	Prerequisiti	309
	Scrivere l'estensione	310
	Uso dell'estensione	319
	Ulteriori letture	320
	Aggiungere un dominio di riferimento	320
	Panoramica	320
	Prerequisiti	321
	Scrivere l'estensione	321
	Uso dell'estensione	330
	Ulteriori letture	331
	Sviluppo di estensioni autodoc	331
	Panoramica	331
	Prerequisiti	332
	Scrivere l'estensione	332
	Uso dell'estensione	334
	Ulteriori letture	335
5.2	Gli How-to	335
	Dipendenza da un'altra estensione	335
	Configurazione dei builder	335
	Evidenziare l'entry point dei builder	335
5.3	Sviluppo di temi HTML	336
	Creazione di temi	336
	Configurazione del tema (<code>theme.toml</code>)	337
	Configurazione del tema (<code>theme.conf</code>)	338
	Converte <code>theme.conf</code> in <code>theme.toml</code>	339
	Distribuire il tema come pacchetto Python	339
	I template	340
	I template	340
	Template statici	348
	Uso di metadati di pagina personalizzati nei template HTML	348
	Definizione delle funzioni del template personalizzato	349
	Aggiunta dei file statici alle risorse della build	350
	Iniettare JavaScript in base alla configurazione dell'utente	350
6	Le API di Sphinx	353
6.1	Oggetti importanti	354
6.2	Fasi della Build	355
6.3	Metadati dell'estensione	356
6.4	API utilizzate per scrivere le estensioni	356
	API dell'applicazione	357
	Configurazione dell'estensione	357

Emissione di eventi	373
Informazioni sul runtime di Sphinx	374
Eventi core di Sphinx	375
Controllo della versione di Sphinx	375
L'oggetto Config	375
Il template bridge	376
Eccezioni	377
API di callback degli eventi	377
Panoramica degli eventi core	378
Dettagli sugli eventi core	380
Eventi specifici del builder	385
API del progetto	387
API per la build dell'environment	388
API del Builder	389
Environment Collector API	393
API di markup Docutils	395
Ruoli	395
Direttive	396
API di dominio	399
Dominio Python	408
API del Parser	409
Classi dei nodi Doctree aggiunte da Sphinx	410
Nodi per descrizioni di oggetti specifici del dominio	410
Nuovi costrutti simili ad ammonizioni	413
Altri nodi a livello di paragrafo	414
Nuovi nodi in linea	414
Nodi speciali	415
API per i Log	416
API i18n	418
Internazionalizzazione dell'estensione (i18n) e localizzazione (l10n) utilizzando l'API i18n	419
Utilità	421
Classi base per i componenti	421
Componenti di utilità	426
Funzioni di utilità	427
Tipi di utility	428
API deprecate	428
7 Personalizzazione di LaTeX	451
7.1 L'impostazione di configurazione latex_elements	452
7.2 La configurazione di sphinxsetup	460
7.3 Chiavi aggiuntive 'sphinxsetup' simili a CSS	470
7.4 Macro e ambienti LaTeX	476
Le Macro	476
Il comando \sphinxbox	480
Ambienti	481
Miscellanea	483

8	Ottenere un supporto	485
9	Contribuire a Sphinx	487
9.1	Contribuire a Sphinx	487
	Richieste d'aiuto	487
	Bug Report e Richieste di Funzionalità	488
	Contributi al codice	488
	Per iniziare	488
	Stile della codifica	490
	Le unit test	490
	Contribuire alla documentazione	492
	Build della documentazione	492
	Traduzioni	492
	Suggerimenti per il debug	493
9.2	Il processo di rilascio di Sphinx	494
	Controllo delle versioni	494
	Deprecare una funzionalità	494
	Politica sulla deprecazione	495
	Warning di deprecazione	495
	Politica per il supporto della versione Python	495
	Procedure di rilascio	496
9.3	Organizzazione del progetto Sphinx	496
	Sviluppatori principali	496
	Linee guida	496
	Appartenenza	497
	Altri collaboratori	497
9.4	Codice di Condotta di Sphinx	497
10	FAQ su Sphinx	499
10.1	Come si fa per...	499
10.2	Usare Sphinx con...	500
10.3	Sphinx e Docutils	502
10.4	Info Epub	503
10.5	Info su Texinfo	504
	Visualizzazione dei Link	504
	Note	505
11	Autori di Sphinx	507
11.1	Manutentori	507
11.2	Contributori	507
11.3	Ex manutentori	510
12	Strumenti da riga di comando	511
12.1	Applicazioni Principali	511
	sphinx-quickstart	511
	Sinossi	511
	Descrizione	511
	Opzioni	511
	Vedere anche	514

	sphinx-build	514
	Sinossi	514
	Descrizione	514
	Opzioni	514
	Variabili d'Ambiente	518
	Deprecazione dei Warning	519
	Vedere anche	520
12.2	Applicazioni Aggiuntive	520
	sphinx-apidoc	520
	Sinossi	520
	Descrizione	520
	Opzioni	521
	Environment	523
	Vedere anche	523
	sphinx-autogen	523
	Sinossi	523
	Descrizione	523
	Opzioni	523
	Esempio	524
	Vedere anche	525

13 Glossario 527

14 Changelog 531

14.1	Release 7.4.0 (in sviluppo)	531
14.2	Release 7.3.7 (released Apr 19, 2024)	533
14.3	Release 7.3.6 (released Apr 17, 2024)	534
14.4	Release 7.3.5 (released Apr 17, 2024)	534
14.5	Release 7.3.4 (released Apr 17, 2024)	534
14.6	Release 7.3.3 (released Apr 17, 2024)	534
14.7	Release 7.3.2 (released Apr 17, 2024)	535
14.8	Release 7.3.1 (released Apr 17, 2024)	535
14.9	Release 7.3.0 (released Apr 16, 2024)	535
14.10	Release 7.2.6 (released Sep 13, 2023)	540
14.11	Release 7.2.5 (released Aug 30, 2023)	541
14.12	Release 7.2.4 (released Aug 28, 2023)	541
14.13	Release 7.2.3 (released Aug 23, 2023)	542
14.14	Release 7.2.2 (released Aug 17, 2023)	542
14.15	Release 7.2.1 (released Aug 17, 2023)	543
14.16	Release 7.2.0 (released Aug 17, 2023)	543
14.17	Release 7.1.2 (released Aug 02, 2023)	546
14.18	Release 7.1.1 (released Jul 27, 2023)	546
14.19	Release 7.1.0 (released Jul 24, 2023)	546
14.20	Release 7.0.1 (released May 12, 2023)	549
14.21	Release 7.0.0 (released Apr 29, 2023)	549
14.22	Release 6.2.1 (released Apr 25, 2023)	550
14.23	Release 6.2.0 (released Apr 23, 2023)	550
14.24	Release 6.1.3 (released Jan 10, 2023)	553

14.25	Release 6.1.2 (released Jan 07, 2023)	554
14.26	Release 6.1.1 (released Jan 05, 2023)	554
14.27	Release 6.1.0 (released Jan 05, 2023)	554
14.28	Release 6.0.1 (released Jan 05, 2023)	556
14.29	Release 6.0.0 (released Dec 29, 2022)	556
14.30	Release 5.3.0 (released Oct 16, 2022)	558
14.31	Release 5.2.3 (released Sep 30, 2022)	558
14.32	Release 5.2.2 (released Sep 27, 2022)	559
14.33	Release 5.2.1 (released Sep 25, 2022)	559
14.34	Release 5.2.0.post0 (released Sep 24, 2022)	559
14.35	Release 5.2.0 (released Sep 24, 2022)	559
14.36	Release 5.1.1 (released Jul 26, 2022)	561
14.37	Release 5.1.0 (released Jul 24, 2022)	561
14.38	Release 5.0.2 (released Jun 17, 2022)	563
14.39	Release 5.0.1 (released Jun 03, 2022)	564
14.40	Release 5.0.0 (released May 30, 2022)	564
14.41	Release 4.5.0 (released Mar 28, 2022)	569
14.42	Release 4.4.0 (released Jan 17, 2022)	571
14.43	Release 4.3.2 (released Dec 19, 2021)	574
14.44	Release 4.3.1 (released Nov 28, 2021)	574
14.45	Release 4.3.0 (released Nov 11, 2021)	575
14.46	Release 4.2.0 (released Sep 12, 2021)	578
14.47	Release 4.1.2 (released Jul 27, 2021)	580
14.48	Release 4.1.1 (released Jul 15, 2021)	581
14.49	Release 4.1.0 (released Jul 12, 2021)	581
14.50	Release 4.0.3 (released Jul 05, 2021)	585
14.51	Release 4.0.2 (released May 20, 2021)	585
14.52	Release 4.0.1 (released May 11, 2021)	586
14.53	Release 4.0.0 (released May 09, 2021)	587
14.54	Release 3.5.5 (in development)	592
14.55	Release 3.5.4 (released Apr 11, 2021)	592
14.56	Release 3.5.3 (released Mar 20, 2021)	593
14.57	Release 3.5.2 (released Mar 06, 2021)	593
14.58	Release 3.5.1 (released Feb 16, 2021)	593
14.59	Release 3.5.0 (released Feb 14, 2021)	594
14.60	Release 3.4.3 (released Jan 08, 2021)	599
14.61	Release 3.4.2 (released Jan 04, 2021)	600
14.62	Release 3.4.1 (released Dec 25, 2020)	600
14.63	Release 3.4.0 (released Dec 20, 2020)	600
14.64	Release 3.3.1 (released Nov 12, 2020)	604
14.65	Release 3.3.0 (released Nov 02, 2020)	604
14.66	Release 3.2.1 (released Aug 14, 2020)	607
14.67	Release 3.2.0 (released Aug 08, 2020)	608
14.68	Release 3.1.2 (released Jul 05, 2020)	612
14.69	Release 3.1.1 (released Jun 14, 2020)	613
14.70	Release 3.1.0 (released Jun 08, 2020)	613
14.71	Release 3.0.4 (released May 27, 2020)	619
14.72	Release 3.0.3 (released Apr 26, 2020)	620

14.73	Release 3.0.2 (released Apr 19, 2020)	620
14.74	Release 3.0.1 (released Apr 11, 2020)	621
14.75	Release 3.0.0 (released Apr 06, 2020)	621
14.76	Release 2.4.5 (released Nov 18, 2021)	628
14.77	Release 2.4.4 (released Mar 05, 2020)	628
14.78	Release 2.4.3 (released Feb 22, 2020)	628
14.79	Release 2.4.2 (released Feb 19, 2020)	629
14.80	Release 2.4.1 (released Feb 11, 2020)	629
14.81	Release 2.4.0 (released Feb 09, 2020)	630
14.82	Release 2.3.1 (released Dec 22, 2019)	633
14.83	Release 2.3.0 (released Dec 15, 2019)	633
14.84	Release 2.2.2 (released Dec 03, 2019)	637
14.85	Release 2.2.1 (released Oct 26, 2019)	637
14.86	Release 2.2.0 (released Aug 19, 2019)	637
14.87	Release 2.1.2 (released Jun 19, 2019)	640
14.88	Release 2.1.1 (released Jun 10, 2019)	640
14.89	Release 2.1.0 (released Jun 02, 2019)	641
14.90	Release 2.0.1 (released Apr 08, 2019)	645
14.91	Release 2.0.0 (released Mar 29, 2019)	646
14.92	Release 1.8.6 (released Nov 18, 2021)	654
14.93	Release 1.8.5 (released Mar 10, 2019)	654
14.94	Release 1.8.4 (released Feb 03, 2019)	655
14.95	Release 1.8.3 (released Dec 26, 2018)	656
14.96	Release 1.8.2 (released Nov 11, 2018)	657
14.97	Release 1.8.1 (released Sep 22, 2018)	659
14.98	Release 1.8.0 (released Sep 13, 2018)	660
14.99	Release 1.7.9 (released Sep 05, 2018)	669
14.100	Release 1.7.8 (released Aug 29, 2018)	669
14.101	Release 1.7.7 (released Aug 19, 2018)	670
14.102	Release 1.7.6 (released Jul 17, 2018)	671
14.103	Release 1.7.5 (released May 29, 2018)	672
14.104	Release 1.7.4 (released Apr 25, 2018)	674
14.105	Release 1.7.3 (released Apr 23, 2018)	674
14.106	Release 1.7.2 (released Mar 21, 2018)	675
14.107	Release 1.7.1 (released Feb 23, 2018)	676
14.108	Release 1.7.0 (released Feb 12, 2018)	678
14.109	Release 1.6.7 (released Feb 04, 2018)	684
14.110	Release 1.6.6 (released Jan 08, 2018)	685
14.111	Release 1.6.5 (released Oct 23, 2017)	686
14.112	Release 1.6.4 (released Sep 26, 2017)	688
14.113	Release 1.6.3 (released Jul 02, 2017)	689
14.114	Release 1.6.2 (released May 28, 2017)	690
14.115	Release 1.6.1 (released May 16, 2017)	692
14.116	Release 1.6 (unreleased)	700
14.117	Release 1.5.6 (released May 15, 2017)	700
14.118	Release 1.5.5 (released Apr 03, 2017)	701
14.119	Release 1.5.4 (released Apr 02, 2017)	701
14.120	Release 1.5.3 (released Feb 26, 2017)	702

14.121	Release 1.5.2 (released Jan 22, 2017)	704
14.122	Release 1.5.1 (released Dec 13, 2016)	706
14.123	Release 1.5 (released Dec 5, 2016)	707
14.124	Release 1.4.9 (released Nov 23, 2016)	716
14.125	Release 1.4.8 (released Oct 1, 2016)	717
14.126	Release 1.4.7 (released Oct 1, 2016)	717
14.127	Release 1.4.6 (released Aug 20, 2016)	718
14.128	Release 1.4.5 (released Jul 13, 2016)	719
14.129	Release 1.4.4 (released Jun 12, 2016)	721
14.130	Release 1.4.3 (released Jun 5, 2016)	722
14.131	Release 1.4.2 (released May 29, 2016)	723
14.132	Release 1.4.1 (released Apr 12, 2016)	726
14.133	Release 1.4 (released Mar 28, 2016)	727
14.134	Release 1.3.6 (released Feb 29, 2016)	733
14.135	Release 1.3.5 (released Jan 24, 2016)	734
14.136	Release 1.3.4 (released Jan 12, 2016)	735
14.137	Release 1.3.3 (released Dec 2, 2015)	737
14.138	Release 1.3.2 (released Nov 29, 2015)	738
14.139	Release 1.3.1 (released Mar 17, 2015)	741
14.140	Release 1.3 (released Mar 10, 2015)	741
14.141	Release 1.3b3 (released Feb 24, 2015)	743
14.142	Release 1.3b2 (released Dec 5, 2014)	745
14.143	Release 1.3b1 (released Oct 10, 2014)	746
14.144	Release 1.2.3 (released Sep 1, 2014)	752
14.145	Release 1.2.2 (released Mar 2, 2014)	753
14.146	Release 1.2.1 (released Jan 19, 2014)	754
14.147	Release 1.2 (released Dec 10, 2013)	757
14.148	Release 1.2 beta3 (released Oct 3, 2013)	758
14.149	Release 1.2 beta2 (released Sep 17, 2013)	759
14.150	Release 1.2 beta1 (released Mar 31, 2013)	762
14.151	Release 1.1.3 (Mar 10, 2012)	768
14.152	Release 1.1.2 (Nov 1, 2011) -- 1.1.1 is a silly version number anyway!	770
14.153	Release 1.1.1 (Nov 1, 2011)	770
14.154	Release 1.1 (Oct 9, 2011)	770
14.155	Release 1.0.8 (Sep 23, 2011)	774
14.156	Release 1.0.7 (Jan 15, 2011)	775
14.157	Release 1.0.6 (Jan 04, 2011)	776
14.158	Release 1.0.5 (Nov 12, 2010)	777
14.159	Release 1.0.4 (Sep 17, 2010)	777
14.160	Release 1.0.3 (Aug 23, 2010)	778
14.161	Release 1.0.2 (Aug 14, 2010)	778
14.162	Release 1.0.1 (Jul 27, 2010)	779
14.163	Release 1.0 (Jul 23, 2010)	779
14.164	Release 0.6.7 (Jun 05, 2010)	783
14.165	Release 0.6.6 (May 25, 2010)	783
14.166	Release 0.6.5 (Mar 01, 2010)	784
14.167	Release 0.6.4 (Jan 12, 2010)	785
14.168	Release 0.6.3 (Sep 03, 2009)	786

14.169	Release 0.6.2 (Jun 16, 2009)	787
14.170	Release 0.6.1 (Mar 26, 2009)	788
14.171	Release 0.6 (Mar 24, 2009)	789
14.172	Release 0.5.2 (Mar 24, 2009)	792
14.173	Release 0.5.1 (Dec 15, 2008)	793
14.174	Release 0.5 (Nov 23, 2008) -- Birthday release!	794
14.175	Release 0.4.3 (Oct 8, 2008)	797
14.176	Release 0.4.2 (Jul 29, 2008)	798
14.177	Release 0.4.1 (Jul 5, 2008)	798
14.178	Release 0.4 (Jun 23, 2008)	799
14.179	Release 0.3 (May 6, 2008)	801
14.180	Release 0.2 (Apr 27, 2008)	802
14.181	Release 0.1.61950 (Mar 26, 2008)	804
14.182	Release 0.1.61945 (Mar 26, 2008)	804
14.183	Release 0.1.61843 (Mar 24, 2008)	804
14.184	Release 0.1.61798 (Mar 23, 2008)	805
14.185	Release 0.1.61611 (Mar 21, 2008)	805
15	Progetti che utilizzano Sphinx	807
15.1	Documentazione che utilizza il tema alabaster	807
15.2	Documentazione che utilizza il tema classic	809
15.3	Documentazione che utilizza il tema sphinxdoc	812
15.4	Documentazione che utilizza il tema nature	813
15.5	Documentazione che utilizza un altro tema nativo	813
15.6	Documentazione che utilizza sphinx_rtd_theme	814
15.7	Documentazione che utilizza sphinx_bootstrap_theme	821
15.8	Documentazione che utilizza pydata_sphinx_theme	822
15.9	Documentazione che utilizza un tema personalizzato o integrato in un sito web	823
15.10	Home page e altri siti non di documentazione	826
15.11	Libri prodotti utilizzando Sphinx	827
15.12	Tesi prodotte utilizzando Sphinx	828
15.13	Progetti che integrano la funzionalità di Sphinx	829
	Indice del modulo Python	831
	Indice	833

CAPITOLO 1

Installazione di Sphinx

Sphinx è un'applicazione Python. Può essere installato in uno dei modi descritti di seguito.

Metodi di installazione

- *Pacchetto PyPI*
- *Pacchetto Conda*
- *Package manager specifico del sistema operativo*
 - *Linux*
 - *macOS*
 - *Windows*
- *Docker*
- *Installazione dell'ultima versione di sviluppo*
- *Installazione dai sorgenti*

Dopo l'installazione, si può verificare che Sphinx sia disponibile eseguendo:

```
$ sphinx-build --version
```

Questo dovrebbe stampare il numero di versione di Sphinx.

Suggerimento: Per lo sviluppo in locale, in genere si consiglia di installare Sphinx in un

ambiente non globale (utilizzando ad esempio gli ambienti [venv](#)³ o [conda](#)⁴). Ciò consentirà l'utilizzo di versioni sphinx separate ed estensioni di terze parti per ciascun progetto Sphinx.

1.1 Pacchetto PyPI

I pacchetti Sphinx vengono pubblicati sul [Python Package Index](#)⁵ (PyPI). Lo strumento preferito per installare pacchetti da PyPI è **pip**, incluso in tutte le moderne versioni di Python.

Eseguire il seguente comando:

```
$ pip install -U sphinx
```

Suggerimento: Per evitare problemi durante la re-build dell'ambiente, è consigliabile aggiungere le versioni sphinx e le estensioni di terze parti in un file [requirements.txt](#) file⁶:

```
$ pip install -r requirements.txt
```

Oppure, se si scrive la documentazione per un pacchetto Python, si inseriscono le dipendenze nel file [pyproject.toml](#)⁷:

```
$ pip install .[docs]
```

1.2 Pacchetto Conda

Per lavorare con **conda**, è necessaria una distribuzione Python basata su conda come [anaconda](#)⁸, [miniconda](#)⁹, [miniforge](#)¹⁰ o [micromamba](#)¹¹.

Sphinx è disponibile sia tramite il canale *anaconda main* (gestito da Anaconda Inc.)

```
$ conda install sphinx
```

sia tramite il canale della community *conda-forge*:

```
$ conda install -c conda-forge sphinx
```

³ <https://docs.python.org/3/library/venv.html>

⁴ <https://conda.io/projects/conda/en/latest/user-guide/getting-started.html>

⁵ <https://pypi.org/project/Sphinx/>

⁶ <https://pip.pypa.io/en/stable/reference/requirements-file-format/>

⁷ <https://packaging.python.org/en/latest/guides/writing-pyproject-toml/#dependencies-optional-dependencies>

⁸ <https://docs.anaconda.com/anaconda/>

⁹ <https://docs.anaconda.com/miniconda/>

¹⁰ <https://github.com/conda-forge/miniforge/>

¹¹ <https://mamba.readthedocs.io/en/latest/installation/micromamba-installation.html>

1.3 Package manager specifico del sistema operativo

Si può installare una versione globale di Sphinx nel sistema utilizzando gestori di pacchetti specifici del sistema operativo. Tuttavia, si tenga presente che questo è meno flessibile e si potrebbero riscontrare problemi di compatibilità se si vuol lavorare su diversi progetti.

Linux

Debian/Ubuntu

Installare `python3-sphinx` usando **apt-get**:

```
$ apt-get install python3-sphinx
```

Se non è già presente, questo installerà anche Python.

RHEL, CentOS

Installare `python-sphinx` usando **yum**:

```
$ yum install python-sphinx
```

Se non è già presente, questo installerà anche Python.

Altre distribuzioni

La maggior parte delle distribuzioni Linux ha Sphinx nei repository dei pacchetti. Di solito il pacchetto si chiama `python3-sphinx`, `python-sphinx` o `sphinx`. Si tenga presente che ci sono almeno altri due pacchetti con `sphinx` nel loro nome: un toolkit di riconoscimento vocale (*CMU Sphinx*) e un database di ricerca full-text (*Sphinx search*).

macOS

Sphinx può essere installato utilizzando [Homebrew](https://brew.sh/)¹², [MacPorts](https://www.macports.org/)¹³.

¹² <https://brew.sh/>

¹³ <https://www.macports.org/>

Homebrew

```
$ brew install sphinx-doc
```

Per ulteriori informazioni, consultare il [package overview](#)¹⁴.

MacPorts

Installare `python3x-sphinx` usando **port**:

```
$ sudo port install py39-sphinx
```

Per impostare i path degli eseguibili, usare il comando `port select`:

```
$ sudo port select --set python python39
$ sudo port select --set sphinx py39-sphinx
```

Per ulteriori informazioni, consultare il [package overview](#)¹⁵.

Windows

Sphinx può essere installato utilizzando [Chocolatey](#)¹⁶.

Chocolatey

```
$ choco install sphinx
```

Si deve [installare Chocolatey](#)¹⁷ prima di eseguirlo.

Per ulteriori informazioni, fare riferimento alla [chocolatey page](#)¹⁸.

1.4 Docker

Le immagini Docker per Sphinx sono pubblicate su [Docker Hub](#)¹⁹. Ci sono due tipi di immagini:

- [sphinxdoc/sphinx](#)²⁰
- [sphinxdoc/sphinx-latexpdf](#)²¹

¹⁴ <https://formulae.brew.sh/formula/sphinx-doc>

¹⁵ <https://www.macports.org/ports.php?by=library&substr=py39-sphinx>

¹⁶ <https://chocolatey.org/>

¹⁷ <https://chocolatey.org/install>

¹⁸ <https://chocolatey.org/packages/sphinx/>

¹⁹ <https://hub.docker.com/>

²⁰ <https://hub.docker.com/r/sphinxdoc/sphinx>

²¹ <https://hub.docker.com/r/sphinxdoc/sphinx-latexpdf>

La prima è per l'utilizzo standard di Sphinx e la seconda viene utilizzata principalmente per le build PDF con LaTeX. Si scelga quella più adatta.

Nota: sphinxdoc/sphinx-latexpdf contiene i pacchetti di TeXLive. L'immagine è molto grande (oltre 2 GB!).

Suggerimento: Con le immagini docker, utilizzare il comando `docker run` per richiamare i comandi sphinx. Ad esempio, si può utilizzare il seguente comando per creare un progetto Sphinx:

```
$ docker run -it --rm -v /path/to/document:/docs sphinxdoc/sphinx_
↳ sphinx-quickstart
```

E si può usare il seguente comando per creare un documento HTML:

```
$ docker run --rm -v /path/to/document:/docs sphinxdoc/sphinx make html
```

Per maggiori dettagli, leggere il [README file](#)²² delle immagini docker.

1.5 Installazione dell'ultima versione di sviluppo

Si può installare l'ultimo sviluppo da *PyPI* usando il flag `--pre`:

```
$ pip install -U --pre sphinx
```

Avvertimento: Generalmente non ce n'è bisogno (o non la si vuole), ma può essere utile se si nota un possibile bug nell'ultima versione stabile.

1.6 Installazione dai sorgenti

Si può installare Sphinx direttamente da un clone del [repository Git](#)²³. Questo può essere fatto clonando il repository e installando dal clone locale, o semplicemente installando direttamente tramite **git**.

```
$ git clone https://github.com/sphinx-doc/sphinx
$ cd sphinx
$ pip install .
```

²² <https://hub.docker.com/r/sphinxdoc/sphinx>

²³ <https://github.com/sphinx-doc/sphinx>

```
$ pip install git+https://github.com/sphinx-doc/sphinx
```

Si può anche scaricare uno snapshot del repository Git in formato [tar.gz](#)²⁴ o [zip](#)²⁵. Una volta scaricati ed estratti, questi possono essere installati con **pip** come sopra.

²⁴ <https://github.com/sphinx-doc/sphinx/archive/master.tar.gz>

²⁵ <https://github.com/sphinx-doc/sphinx/archive/master.zip>

CAPITOLO 2

Per Iniziare

Sphinx è un *generatore di documentazione* ovvero uno strumento che traduce un insieme di file di testo in vari formati di output, producendo automaticamente riferimenti incrociati, indici, ecc. Cioè, se si ha una directory contenente un mucchio di documenti *reStructuredText* o *Markdown*, Sphinx può generare una serie di file HTML, un file PDF (tramite LaTeX), pagine man e molto altro.

Sphinx si concentra sulla documentazione, in particolare sulla documentazione scritta a mano, tuttavia, Sphinx può essere utilizzato anche per generare blog, home page e persino libri. Gran parte della potenza di Sphinx deriva dalla ricchezza del suo formato di markup di default di semplice testo, *reStructuredText*, insieme alle sue *significant extensibility capabilities*.

L'obiettivo di questo documento è quello di fornire un rapido assaggio di cos'è Sphinx e di come si possa usare. Dopo questa lettura, si può dare un'occhiata alla *installation guide* seguita dall'introduzione al formato di markup di default utilizzato da Sphinx, *reStructuredText*.

Per una grande "introduzione" alla scrittura di documenti in generale -- i perché e i come, si può consultare *Write the docs*²⁶, scritto da Eric Holscher.

2.1 Impostazione dei sorgenti della documentazione

La directory root di una collezione Sphinx di documenti sorgenti in formato testo è detta *source directory*. Questa directory contiene anche il file di configurazione di Sphinx `conf.py`, dove si possono configurare tutti gli aspetti di come Sphinx legge i sorgenti e costruisce la documentazione.²⁹

²⁶ <https://www.writethedocs.org/guide/writing/beginners-guide-to-docs/>

²⁹ Questo è il layout solito. Tuttavia, `conf.py` può anche risiedere in un'altra directory, la *directory di configurazione*. Fare riferimento alla *sphinx-build man page* per maggiori informazioni.

Sphinx viene fornito con uno script chiamato **sphinx-quickstart** che imposta una directory sorgente e crea un `conf.py` di default con i valori di configurazione più utili derivanti dalle poche domande che pone. Per usarlo, si esegue:

```
$ sphinx-quickstart
```

2.2 Definizione della struttura del documento

Supponiamo che sia stato eseguito **sphinx-quickstart**. Esso ha creato una directory sorgente con `conf.py` e un "documento root", `index.rst`. La funzione principale del *root document* [documento root] è quella di fungere da pagina di benvenuto e di contenere la radice della "struttura del sommario" (o *toctree*). Questa è una delle cose principali che Sphinx aggiunge a reStructuredText, un modo per collegare più file a un'unica gerarchia di documenti.

Direttive reStructuredText

toctree è una *direttiva* di reStructuredText, un markup molto versatile. Le direttive possono avere argomenti, opzioni e contenuto.

Gli *argumenti* vengono dati subito dopo i due punti dopo il nome della direttiva. Ogni direttiva decide se può avere argomenti e quanti.

Le *opzioni* vengono dopo gli argomenti, sotto forma di un "elenco di campi". `maxdepth` è un'opzione di questo tipo per la direttiva *toctree*.

Il *contenuto* segue le opzioni o gli argomenti dopo una riga vuota. Ogni direttiva decide se consentire il contenuto e cosa farne.

Una pratica comune con le direttive è che **la prima riga del contenuto deve essere rientrata allo stesso livello delle opzioni**.

La direttiva *toctree* inizialmente è vuota e si presenta così:

```
.. toctree::  
   :maxdepth: 2
```

L'elenco dei documenti va aggiunto nel *contenuto* della direttiva:

```
.. toctree::  
   :maxdepth: 2  
  
   usage/installation  
   usage/quickstart  
   ...
```

Questo è esattamente l'aspetto del *toctree* per questa documentazione. I documenti da includere sono dati come i *document name*, che in pratica significa che si tralascia l'estensione del nome del file e si usano le barre (/) come separatori di directory.

Vedi anche:

Maggiori informazioni *sulla direttiva toctree*.

Ora si possono creare i file che elencati nel `toctree` e si possono aggiungere i contenuti, i loro titoli di sezione verranno inseriti (fino al livello `maxdepth`) nel punto in cui il viene inserita la direttiva `toctree`. Inoltre, Sphinx ora conosce l'ordine e la gerarchia dei documenti. (Possono contenere essi stessi direttive `toctree`, il che significa che si possono creare gerarchie profondamente nidificate se necessario).

2.3 Aggiungere il contenuto

Nei file sorgenti di Sphinx, si possono utilizzare la maggior parte delle funzioni dello standard *reStructuredText*. Ci sono anche diverse funzionalità aggiunte da Sphinx. Ad esempio, si possono aggiungere riferimenti tra file in modo portabile (funzionanti per tutti i tipi di output) utilizzando il ruolo *ref*.

Ad esempio, se si sta visualizzando la versione HTML, si può guardare il sorgente del documento -- usando il link "Show Source" [Mostra sorgente] nella barra laterale.

Da fare: Aggiornare il link seguente quando verranno aggiunte le nuove guide a riguardo.

Vedi anche:

reStructuredText per un'introduzione più approfondita a *reStructuredText*, inclusi i markup aggiunti da Sphinx.

2.4 Esecuzione della build

Dopo aver aggiunto alcuni file e contenuti, creiamo una prima build dei documenti. Una build viene avviata con il programma **sphinx-build**:

```
$ sphinx-build -M html sourcedir outputdir
```

dove *sourcedir* è la *source directory* e *outputdir* è la directory in cui si vuol mettere la documentazione. L'opzione *-M* seleziona un builder; in questo esempio Sphinx creerà file HTML.

Vedi anche:

Fare riferimento a *sphinx-build man page* per tutte le opzioni supportate da **sphinx-build**.

Tuttavia, lo script **sphinx-quickstart** crea un `Makefile` e un `make.bat` che facilitano ulteriormente la vita. Questi si possono eseguire col comando **make** e il nome del builder. Per esempio.

```
$ make html
```

Questo creerà documenti HTML nella directory di build scelta. Eseguire **make** senza un argomento per vedere quali 'target' sono disponibili.

Come si generano i documenti PDF?

`make latexpdf` esegue il *builder LaTeX* e invoca subito la toolchain pdfTeX.

Da fare: Spostare l'intera sezione in una guida su rST o sulle direttive

2.5 Documentare gli oggetti

Uno degli obiettivi principali di Sphinx è quello di facilitare la documentazione degli *oggetti* (nel senso più generale) in qualsiasi *dominio*. Un dominio è una raccolta di tipi di oggetti che appartengono ad un insieme, completi di markup per creare e fare riferimento alle loro descrizioni.

Il dominio più importante è quello di Python. Ad esempio, per documentare la funzione nativa di Python `enumerate()`, la si dovrebbe aggiungere a uno dei file sorgenti.

```
.. py:function:: enumerate(sequence[, start=0])
```

Return an iterator that yields tuples of an index and an item of the **sequence**. (And so on.)

Questo è reso così:

```
enumerate(sequence[, start=0])
```

Restituisce un iteratore che produce tuple di un indice e un elemento di *sequence*. (E così via).

L'argomento della direttiva è la *signature* [firma] dell'oggetto descritto, il contenuto ne è la documentazione. Possono esserci più firme, ciascuna nella propria riga.

Il dominio Python è anche quello di default, quindi non è necessario anteporre il prefisso al markup con il nome del dominio.

```
.. function:: enumerate(sequence[, start=0])
```

```
...
```

fa la stessa cosa se si mantiene l'impostazione di default per il dominio.

Ci sono molte altre direttive per documentare altri tipi di oggetti Python, per esempio *py:class* e *py:method*. C'è anche un riferimento incrociato *role* per ciascuno di questi tipi di oggetto. Questo markup creerà un link alla documentazione di `enumerate()`.

The `:py:func:`enumerate`` function can be used for ...

E questa è la prova: Un link a [`enumerate\(\)`](#).

Di nuovo, `py:` può essere omissso se il dominio Python è quello di default. Non importa quale file contenga la documentazione effettiva per `enumerate()`; Sphinx lo troverà e creerà un link ad essa.

Ogni dominio avrà regole speciali su come possono apparire le firme e renderà l'output formattato in modo particolare o aggiungerà funzionalità specifiche come link a tipi di parametri, ad es. nei domini C/C++.

Vedi anche:

[*Domini*](#) per tutti i domini disponibili e le loro direttive/ruoli.

2.6 Configurazione di base

In precedenza abbiamo accennato al fatto che il file `conf.py` controlla il modo in cui Sphinx elabora i documenti. In tale file, che viene eseguito come sorgente Python, si assegnano i valori di configurazione. Per utenti esperti: poiché viene eseguito da Sphinx, si possono eseguire attività non banali, come estendere `sys.path`²⁷ o importare un modulo per cercare la versione che si sta documentando.

I valori di configurazione che probabilmente si vorranno modificare sono già inseriti in `conf.py` da **sphinx-quickstart** e inizialmente commentati (con la sintassi standard di Python: un `#` commenta il resto della riga). Per modificare il valore di default, rimuovere il cancelletto e modificare il valore. Per personalizzare una configurazione che non viene aggiunta automaticamente da **sphinx-quickstart**, basta aggiungere un'ulteriore assegnazione.

Si tenga presente che il file utilizza la sintassi Python per stringhe, numeri, elenchi ecc. Il file viene salvato in UTF-8 per default, come indicato dalla dichiarazione della codifica nella prima riga.

Vedi anche:

[*Configurazione*](#) per la documentazione di tutti i valori di configurazione disponibili.

Da fare: Spostare l'intero documento in un'altra sezione

²⁷ <https://docs.python.org/3/library/sys.html#sys.path>

2.7 Autodoc

Quando si documenta il codice Python, si è soliti inserire molta documentazione nei sorgenti, nelle stringhe di documentazione. Sphinx supporta l'inclusione di docstring dai moduli con una *extension* (una *extension* è un modulo Python che fornisce funzionalità aggiuntive per i progetti Sphinx) chiamata *autodoc*.

Vedi anche:

[*sphinx.ext.autodoc*](#) per la descrizione completa delle funzionalità di autodoc.

2.8 Intersphinx

Molti documenti Sphinx, inclusa la [documentazione Python](#)²⁸, sono pubblicati su Internet. Quando si vogliono creare collegamenti a tali documenti nella documentazione, lo si fa con [*sphinx.ext.intersphinx*](#).

Per usare intersphinx, lo si deve attivare in `conf.py` inserendo la stringa `'sphinx.ext.intersphinx'` in `extensions` e impostare il valore di configurazione `intersphinx_mapping`.

Ad esempio, per avere un link a `io.open()` nel manuale della libreria Python, si deve impostare `intersphinx_mapping` come:

```
intersphinx_mapping = {'python': ('https://docs.python.org/3', None)}
```

E ora si può scrivere un riferimento incrociato come `:py:func:`io.open``. Qualsiasi riferimento incrociato che non ha un target corrispondente nel set di documentazione corrente, verrà cercato nei set di documentazione configurati in `intersphinx_mapping` (questo richiede l'accesso all'URL per scaricare l'elenco dei target validi). Intersphinx funziona anche per altri ruoli di *domain* incluso `:ref:`, tuttavia non funziona per `:doc:` poiché è un ruolo non-dominio.

Vedi anche:

[*sphinx.ext.intersphinx*](#) per la descrizione completa delle funzionalità di intersphinx.

2.9 Altri argomenti da trattare

- *Altre estensioni:*
- File statici
- *Selezione di un tema*
- *Templating*
- Utilizzo delle estensioni

²⁸ <https://docs.python.org/3>

- *Scrittura di estensioni*

CAPITOLO 3

La build del primo progetto

In questo tutorial creeremo un semplice progetto di documentazione utilizzando Sphinx e la visualizzeremo nel browser come HTML. Il progetto includerà narrativa, documentazione scritta a mano e documentazione di API generata automaticamente.

Il tutorial è rivolto ai nuovi arrivati che desiderano apprendere i fondamentali di come vengono creati e strutturati i progetti. Creeremo una libreria software fittizia per generare ricette alimentari casuali che serviranno da guida durante tutto il processo, con l'obiettivo di documentarla opportunamente.

Per mostrare le funzionalità di Sphinx per la documentazione del codice utilizzeremo Python, che supporta anche la generazione di documentazione *automatic*.

Nota: Molti altri linguaggi sono supportati nativamente in Sphinx per la documentazione del codice *manuale*, tuttavia sono richieste delle estensioni per la documentazione *automatica*, come [Breathe](https://breathe.readthedocs.io/)³⁰.

Per seguire le istruzioni avremo bisogno dell'accesso a una riga di comando simile a quella di Linux e di una conoscenza di base del suo funzionamento, nonché di un'installazione Python funzionante per lo sviluppo, poiché utilizzeremo *ambienti virtuali Python* per creare il progetto.

³⁰ <https://breathe.readthedocs.io/>

3.1 Per iniziare

Configurazione del progetto e dell'ambiente di sviluppo

In una nuova directory, si crea un file chiamato `README.rst` con il seguente contenuto.

Listato 1: `README.rst`

```
Lumache
=====

**Lumache** (/lu'make/) is a Python library for cooks and food lovers.
→ that
creates recipes mixing random ingredients.
```

È un buon momento per creare un ambiente virtuale Python e installare gli strumenti richiesti. Per farlo, si apre un terminale a riga di comando, `cd` nella directory che appena creata, e si eseguono i seguenti comandi:

```
$ python -m venv .venv
$ source .venv/bin/activate
(.venv) $ python -m pip install sphinx
```

Nota: Il metodo di installazione utilizzato sopra è descritto più dettagliatamente in [Pacchetto PyPI](#). Per il resto di questo tutorial, le istruzioni presupporranno un ambiente virtuale Python.

Eseguendo correttamente queste istruzioni, si avranno a disposizione gli strumenti a riga di comando di Sphinx. Si può eseguire una verifica di base eseguendo questo comando:

```
(.venv) $ sphinx-build --version
sphinx-build 4.0.2
```

Se si vede un output simile, si è sulla strada giusta!

Creazione del layout della documentazione

Dalla riga di comando, poi, si esegue il comando seguente:

```
(.venv) $ sphinx-quickstart docs
```

Questo presenterà una serie di domande necessarie per creare la directory di base e il layout della configurazione del progetto all'interno della cartella `docs`. Per procedere, rispondere ad ogni domanda come segue:

- `> Separate source and build directories (y/n) [n]:` Scrivere "y" (senza virgolette) e premere Invio.

- > Project name: Scrivere "Lumache" (senza virgolette) e premere Invio.
- > Author name(s): Scrivere "Graziella" (senza virgolette) e premere Invio.
- > Project release []: Scrivere "0.1" (senza virgolette) e premere Invio.
- > Project language [en]: Lasciarlo vuoto (il default è English) e premere Invio.

Dopo l'ultima domanda, si vedrà la nuova directory docs con il seguente contenuto.

```
docs
├── build
├── make.bat
├── Makefile
├── source
│   ├── conf.py
│   ├── index.rst
│   ├── _static
│   └── _templates
```

Lo scopo di ciascuno di questi file è:

build/

Una directory vuota (per ora) che conterrà la documentazione renderizzata.

make.bat e Makefile

Script utili per semplificare alcune operazioni comuni di Sphinx, come il rendering del contenuto.

source/conf.py

Uno script Python che contiene la configurazione del progetto Sphinx. Contiene il nome del progetto e la versione specificata in `sphinx-quickstart`, oltre ad alcune voci di configurazioni aggiuntive.

source/index.rst

Il *root document* del progetto, che funge da pagina di benvenuto e contiene il "sommario" principale (o *toctree*).

Grazie a questo avvio, c'è già tutto il necessario per creare la documentazione in HTML per la prima volta. Per fare ciò, si esegue questo comando:

```
(.venv) $ sphinx-build -M html docs/source/ docs/build/
```

Infine, si apre `docs/build/html/index.html` nel browser. Si dovrebbe vedere qualcosa del genere:

Eccoci qua! Abbiamo creato la prima documentazione HTML utilizzando Sphinx. Ora si può iniziare a *personalizzarlo*.



Fig. 1: Documentazione appena creata di Lumache

3.2 Primi passi per documentare il progetto utilizzando Sphinx

La build della documentazione HTML

Il file `index.rst` creato da `sphinx-quickstart` ha già del contenuto e viene visualizzato come prima pagina della documentazione HTML. È scritto in reStructuredText, un potente linguaggio di markup.

Modificare il file come segue:

Listato 2: `docs/source/index.rst`

```
Welcome to Lumache's documentation!
=====

**Lumache** (/lu'make/) is a Python library for cooks and food lovers.
→ that
creates recipes mixing random ingredients. It pulls data from the
→ `Open Food
Facts database <https://world.openfoodfacts.org/>`_ and offers a
→ simple and
intuitive API.

.. note::

    This project is under active development.
```

Questo mostra diverse funzionalità della sintassi del reStructuredText, tra cui:

- un **titolo di sezione** utilizzando `===` per la sottolineatura,
- due esempi di *Markup in linea*: `**enfasi forte**` (solitamente in grassetto) e `*enfasi*` (di solito in corsivo),
- un **link esterno 'inline'**,
- e una nota **ammonimento** (una delle *direttive* disponibili)

Ora per renderizzarlo col nuovo contenuto, si può utilizzare il comando `sphinx-build` come prima o sfruttare il comodo script come segue:

```
(.venv) $ cd docs
(.venv) $ make html
```

Dopo aver eseguito questo comando, vedremo che `index.html` riflette le nuove modifiche!

Build della documentazione in altri formati

Sphinx supporta una varietà di formati oltre all'HTML, inclusi PDF, EPUB, *e altri*. Ad esempio, per creare la documentazione in formato EPUB, si esegue questo comando dalla directory `docs`:

```
(.venv) $ make epub
```

Dopodiché, si vedranno i file corrispondenti all'e-book in `docs/build/epub/`. Si può aprire `Lumache.epub` con un visualizzatore di e-book compatibile con EPUB, come [Calibre](https://calibre-ebook.com/)³¹, oppure visualizzare l'anteprima `index.xhtml` su un browser web.

Nota: Per visualizzare rapidamente un elenco completo dei possibili formati di output, oltre ad alcuni comandi extra utili, si può eseguire `make help`.

Ogni formato di output ha alcune opzioni di configurazione specifiche ottimizzabili, *incluso EPUB*. Ad esempio, il valore di default di `epub_show_urls` è `inline`, il che significa che, per default, gli URL vengono visualizzati subito dopo il link corrispondente, tra parentesi. Tale comportamento è modificabile aggiungendo il seguente codice alla fine di `conf.py`:

```
# EPUB options
epub_show_urls = 'footnote'
```

Con questa configurazione, e dopo aver eseguito nuovamente `make epub`, si noterà che gli URL ora appaiono come note a piè di pagina, evitando così di ingombrare il testo. Stupendo! Continuare a leggere e a esplorare *altri modi di personalizzare Sphinx*.

Nota: Un PDF si genera con Sphinx eseguendo `make latexpdf`, a condizione che il sistema abbia un'installazione LaTeX funzionante, come spiegato nella documentazione di [sphinx.builders.latex.LaTeXBuilder](#). Sebbene ciò sia perfettamente fattibile, tali installazioni sono spesso grandi e in generale LaTeX richiede in alcuni casi un'attenta configurazione, quindi la generazione di PDF non rientra nell'ambito di questo tutorial.

³¹ <https://calibre-ebook.com/>

3.3 Altre personalizzazioni di Sphinx

Esistono due modi principali per personalizzare la documentazione oltre a ciò che è possibile con il solo Sphinx: estensioni e temi.

Abilitazione di un'estensione nativa

Oltre a questi valori di configurazione, Sphinx si può personalizzare ulteriormente utilizzando le *estensioni*. Sphinx ne fornisce diverse *nativamente* e ce ne sono molte altre *gestite dalla community*.

Ad esempio, per abilitare l'estensione `sphinx.ext.duration`, si individua la lista delle `extensions` in `conf.py` e si aggiunge un elemento come segue:

Listato 3: docs/source/conf.py

```
# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
extensions = [
    'sphinx.ext.duration',
]
```

Successivamente, ogni volta che si genera la documentazione, si vedrà un breve rapporto sulla durata alla fine dell'output nella console, come questo:

```
(.venv) $ make html
...
The HTML pages are in build/html.

===== slowest reading durations_
->=====
0.042 temp/source/index
```

Uso di un tema HTML di terze parti

I temi, invece, rappresentano un modo per personalizzare l'aspetto della documentazione. Sphinx ha diversi *temi nativi* e ce ne sono anche di *terze parti*³².

Ad esempio, per utilizzare il tema di terze parti `Furo`³³ nella documentazione HTML, lo si dovrà prima installare con `pip` nell'ambiente virtuale Python, in questo modo:

```
(.venv) $ pip install furo
```

Poi, si individua la variabile `html_theme` in `conf.py` e se ne sostituisce il valore come segue:

³² <https://sphinx-themes.org/>

³³ <https://pradyunsg.me/furo/>

Listato 4: docs/source/conf.py

```
# The theme to use for HTML and HTML Help pages. See the
# documentation for
# a list of builtin themes.
#
html_theme = 'furo'
```

Con questa modifica, si noterà che la documentazione HTML ha ora un nuovo aspetto:



Fig. 2: Documentazione HTML di Lumache con il tema Furo

È ora il momento di *espandere la documentazione narrativa e suddividerla in diversi documenti*.

3.4 Documentazione narrativa in Sphinx

Strutturare la documentazione su più pagine

Il file `index.rst` creato da `sphinx-quickstart` è il *root document*, la cui funzione principale è quella di servire come pagina iniziale e per contenere il "sommario" principale (o *toctree*). Sphinx consente di assemblare un progetto da diversi file, il che è utile quando il progetto cresce.

Come esempio, creiamo un nuovo file `docs/source/usage.rst` (accanto a `index.rst`) con questi contenuti:

Listato 5: docs/source/usage.rst

```
Usage
=====

Installation
-----

To use Lumache, first install it using pip:

.. code-block:: console
```

(continues on next page)

(continua dalla pagina precedente)

```
(.venv) $ pip install lumache
```

Questo nuovo file contiene due titoli di *sezione*, il testo di un paragrafo normale e una direttiva *code-block* che fa apparire un blocco di contenuto come codice sorgente, con l'appropriata evidenziazione della sintassi (in questo caso, il testo generico *console*).

La struttura del documento è determinata dalla successione degli stili dei titoli, ciò significa che, utilizzando `---` per la sezione "Installazione" dopo `===` per la sezione "Usage" [Utilizzo], si è dichiarato che "Installazione" è una *sottosezione* di "Usage".

Per completare il processo, si aggiunge un *toctree* (sommario) tramite una *direttiva* alla fine di `index.rst` includendo il documento appena creato, come segue:

Listato 6: docs/source/index.rst

```
Contents
```

```
-----
```

```
.. toctree::
```

```
    usage
```

Questo passaggio inserisce il documento all'inizio del *toctree*, quindi ora appartiene alla struttura del progetto, che finora assomiglia a questo:

```
index
```

```
└─ usage
```

Se si crea la documentazione HTML eseguendo `make html`, si vedrà che *toctree* viene visualizzato come un elenco di link ipertestuali e questo consente di navigare verso la pagina appena creata. Semplice!

Avvertimento: I documenti all'esterno di un *toctree* produrranno un messaggio di **WARNING: document isn't included in any toctree** durante il processo di build e saranno irraggiungibili per gli utenti.

Aggiungere i riferimenti incrociati

Una potente caratteristica di Sphinx è la possibilità di aggiungere senza soluzione di continuità *riferimenti incrociati* a parti specifiche della documentazione: un documento, una sezione, una figura, del codice, ecc. Questo tutorial ne è pieno!

Per aggiungere un riferimento incrociato, si scrive questa frase subito dopo il paragrafo introduttivo in `index.rst`:

Listato 7: docs/source/index.rst

Check out the `:doc:`usage`` section for further information.

L'uso di *doc*, *che è un ruolo*, fa automaticamente riferimento a un documento specifico nel progetto, in questo caso `usage.rst` creato in precedenza.

In alternativa, si può anche aggiungere un riferimento incrociato a una parte arbitraria del progetto. Per questo, si deve utilizzare il ruolo *ref* e aggiungere una *etichetta* esplicita che agisca come un *target*³⁴.

Ad esempio, per fare riferimento alla sottosezione "Installazione", si aggiunge un'etichetta subito prima del titolo, come segue:

Listato 8: docs/source/usage.rst

Usage

=====

.. _installation:

Installation

...

E si cambia la frase aggiunta in `index.rst` simile a questa:

³⁴ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#hyperlink-targets>

Listato 9: docs/source/index.rst

```
Check out the :doc:`usage` section for further information, including
→ how to
:ref:`install <installation>` the project.
```

Da notare il trucco: la parte `installare` specifica come apparirà il link (vogliamo che sia una parola specifica, quindi una frase significativa), mentre la parte `<installation>` si riferisce all'etichetta effettiva a cui vogliamo aggiungere un riferimento incrociato. Se non si include un titolo esplicito, quindi si usa `:ref:`installation``, verrà utilizzato il titolo della sezione (in questo caso, Installazione). Entrambi i ruoli `:doc:` e `:ref:` verranno visualizzati come link ipertestuali nella documentazione HTML.

E per *documentare le porzioni di codice in Sphinx*? Continuare a leggere!

3.5 Descrizione del codice in Sphinx

Nelle *sezioni precedenti del tutorial* è descritto come scrivere una documentazione narrativa o in prosa in Sphinx. In questa sezione si descriveranno invece gli oggetti codice.

Sphinx supporta la documentazione di codice in diversi linguaggi, ovvero Python, C, C++, JavaScript e reStructuredText. Ognuno può essere documentato utilizzando una serie di direttive e ruoli raggruppati per *dominio*. Per il resto del tutorial utilizzeremo quello di Python, ma tutti i concetti visti in questa sezione si applicano anche agli altri domini.

Python

Documentare oggetti Python

Sphinx offre diversi ruoli e direttive per documentare gli oggetti Python, tutti raggruppati insieme nel *dominio Python*. Ad esempio, si può utilizzare la direttiva `py:function` per documentare una funzione Python, in questo modo:

Listato 10: docs/source/usage.rst

Creating recipes

```
To retrieve a list of random ingredients,
you can use the ``lumache.get_random_ingredients()`` function:
```

```
.. py:function:: lumache.get_random_ingredients(kind=None)
```

```
    Return a list of random ingredients as strings.
```

(continues on next page)

(continua dalla pagina precedente)

```

:param kind: Optional "kind" of ingredients.
:type kind: list[str] or None
:return: The ingredients list.
:rtype: list[str]

```

Questo è reso così:

Creating recipes

To retrieve a list of random ingredients, you can use the `lumache.get_random_ingredients()` function:

`lumache.get_random_ingredients(kind=None)`

Return a list of random ingredients as strings.

PARAMETERS

kind (*list[str] or None*) – Optional “kind” of ingredients.

RETURNS

The ingredients list.

RETURN TYPE

list[str]

Fig. 3: Il risultato della documentazione di una funzione Python in Sphinx

Diverse cose da notare:

- Sphinx ha "parserizzato" l'argomento della direttiva `.. py:function` ed evidenziato il modulo, il nome della funzione e i parametri in modo appropriato.
- Il contenuto della direttiva include una descrizione mono-riga della funzione, nonché un *elenco di campi info* contenente il parametro della funzione, il tipo previsto, il valore restituito e il tipo restituito.

Nota: Il prefisso `py:` specifica il *domain* (dominio). Si può configurare il dominio di default in modo da omettere il prefisso, sia a livello globale usando la configurazione `primary_domain`, sia usando la direttiva `default-domain` per cambiarlo dal punto in cui viene chiamato fino al fine del file. Ad esempio, se lo si imposta su `py` (il default), si può scrivere `.. function::` direttamente.

Riferimenti incrociati di oggetti Python

Per default, la maggior parte di queste direttive generano entità a cui è possibile fare un riferimento incrociato da qualsiasi parte della documentazione utilizzando *un ruolo corrispondente*. Nel caso delle funzioni, si può usare `py:func`, in questo modo:

Listato 11: docs/source/usage.rst

```
The ``kind`` parameter should be either ``"meat"`` , ``"fish"`` ,  
or ``"veggies"``. Otherwise, :py:func:`lumache.get_random_ingredients`  
will raise an exception.
```

Quando si genera la documentazione del codice, Sphinx genererà automaticamente un riferimento incrociato semplicemente utilizzando il nome dell'oggetto, senza che si debba utilizzare esplicitamente un ruolo. Ad esempio, si può descrivere l'eccezione custom sollevata dalla funzione utilizzando la direttiva `py:exception`:

Listato 12: docs/source/usage.rst

```
.. py:exception:: lumache.InvalidKindError  
  
    Raised if the kind is invalid.
```

Poi, si aggiunge questa eccezione alla descrizione originale della funzione:

Listato 13: docs/source/usage.rst

```
.. py:function:: lumache.get_random_ingredients(kind=None)  
  
    Return a list of random ingredients as strings.  
  
    :param kind: Optional "kind" of ingredients.  
    :type kind: list[str] or None  
    :raise lumache.InvalidKindError: If the kind is invalid.  
    :return: The ingredients list.  
    :rtype: list[str]
```

E infine, ecco come apparirebbe il risultato:

Bello, vero?

Creating recipes

To retrieve a list of random ingredients, you can use the `lumache.get_random_ingredients()` function:

```
lumache.get_random_ingredients(kind=None)
    Return a list of random ingredients as strings.

PARAMETERS
    kind (list[str] or None) – Optional “kind” of ingredients.

RAISES
    lumache.InvalidKindError – If the kind is invalid.

RETURNS
    The ingredients list.

RETURN TYPE
    list[str]

The kind parameter should be either "meat", "fish", or "veggies". Otherwise,
lumache.get_random_ingredients() will raise an exception.

exception lumache.InvalidKindError
    Raised if the kind is invalid.
```

Fig. 4: Risultato HTML della documentazione di una funzione Python in Sphinx con riferimenti incrociati

Includere i doctest nella documentazione

Dato che ora si sta descrivendo il codice di una libreria Python, diventerà utile mantenere sia la documentazione che il codice il più sincronizzati possibile. Uno dei modi per farlo in Sphinx è includere frammenti di codice nella documentazione, chiamati *doctest*, che vengono eseguiti quando la documentazione viene creata.

Per mostrare i doctest e le altre funzionalità di Sphinx trattate in questo tutorial, Sphinx dovrà essere in grado di importare il codice. A tale scopo, si scrive questo all'inizio di `conf.py`:

Listato 14: `docs/source/conf.py`

```
# If extensions (or modules to document with autodoc) are in another_
→directory,
# add these directories to sys.path here.
import pathlib
import sys
sys.path.insert(0, pathlib.Path(__file__).parents[2].resolve().as_
→posix())
```

Nota: Un'alternativa alla modifica della variabile `sys.path`³⁵ è quella di creare un file `pyproject.toml` e rendere il codice installabile, in modo che si comporti come qualsiasi altra libreria Python. Tuttavia, l'approccio con `sys.path` è più semplice.

³⁵ <https://docs.python.org/3/library/sys.html#sys.path>

Quindi, prima di aggiungere i doctest alla documentazione, si abilita l'estensione *doctest* in `conf.py`:

Listato 15: docs/source/conf.py

```
extensions = [  
    'sphinx.ext.duration',  
    'sphinx.ext.doctest',  
]
```

Poi, si scrive un blocco doctest in questo modo:

Listato 16: docs/source/usage.rst

```
>>> import lumache  
>>> lumache.get_random_ingredients()  
['shells', 'gorgonzola', 'parsley']
```

I doctest includono le istruzioni Python da eseguire precedute da `>>>`, il prompt standard dell'interprete Python, nonché l'output atteso di ciascuna istruzione. In questo modo Sphinx può verificare se l'output effettivo corrisponde a quello previsto.

Per osservare come appare un errore di doctest (piuttosto che un codice di errore come sopra), scriviamo prima il valore restituito in modo errato. Pertanto, si aggiunge una funzione `get_random_ingredients` come questa:

Listato 17: lumache.py

```
def get_random_ingredients(kind=None):  
    return ["eggs", "bacon", "spam"]
```

Ora si può lanciare `make doctest` per eseguire i doctest della documentazione. Inizialmente verrà visualizzato un errore, poiché il codice effettivo non si comporta come specificato:

```
(.venv) $ make doctest  
Running Sphinx v4.2.0  
loading pickled environment... done  
...  
running tests...  
  
Document: usage  
-----  
*****  
File "usage.rst", line 44, in default  
Failed example:  
    lumache.get_random_ingredients()  
Expected:  
    ['shells', 'gorgonzola', 'parsley']  
Got:  
    ['eggs', 'bacon', 'spam']
```

(continues on next page)

(continua dalla pagina precedente)

```
*****
...
make: *** [Makefile:20: doctest] Error 1
```

Come si può vedere, doctest riporta i risultati attesi e quelli effettivi, per un facile esame. È giunto il momento di correggere la funzione:

Listato 18: lumache.py

```
def get_random_ingredients(kind=None):
    return ["shells", "gorgonzola", "parsley"]
```

E infine, `make doctest` segnala il successo!

Per i grandi progetti, tuttavia, questo approccio manuale può diventare un po' noioso. Nella sezione successiva vedremo *come automatizzare il processo*.

Altri linguaggi (C, C++, altri)

Documentare e cross-referenziare gli oggetti

Sphinx supporta anche la documentazione e i riferimenti incrociati di oggetti scritti in altri linguaggi di programmazione. Sono disponibili nativamente quattro domini aggiuntivi: C, C++, JavaScript e reStructuredText. Le estensioni di terze parti possono definire domini per più linguaggi, come

- Fortran³⁶,
- Julia³⁷, o
- PHP³⁸.

Ad esempio, per documentare una definizione di tipo C++, si deve utilizzare la direttiva `cpp:type`, in questo modo:

```
.. cpp:type:: std::vector<int> CustomList

A typedef-like declaration of a type.
```

Il che darebbe il seguente risultato:

```
typedef std::vector<int> CustomList
```

Una dichiarazione di tipo simile a typedef.

Tutte queste direttive generano poi riferimenti a cui è possibile riferirsi utilizzando il ruolo corrispondente. Ad esempio, per fare riferimento alla definizione di tipo precedente, si può utilizzare il ruolo `cpp:type` in questo modo:

³⁶ <https://sphinx-fortran.readthedocs.io>

³⁷ <https://bastikr.github.io/sphinx-julia>

³⁸ <https://github.com/markstory/sphinxcontrib-phpdomain>

Cross reference to `:cpp:type:`CustomList``.

Ciò produrrebbe un collegamento ipertestuale alla definizione precedente: *CustomList*.

3.6 Generazione automatica della documentazione dal codice

Nella *sezione precedente* del tutorial abbiamo documentato manualmente una funzione Python in Sphinx. Tuttavia, la descrizione non era sincronizzata con il codice stesso, poiché la firma [signature] della funzione non era la stessa. Inoltre, sarebbe carino riutilizzare le **docstring Python**³⁹ nella documentazione, invece di dover scrivere le informazioni in due posti.

Fortunatamente, *l'estensione autodoc* fornisce questa funzionalità.

Riutilizzo di firme e docstring con autodoc

Per utilizzare autodoc, lo si aggiunge prima all'elenco delle estensioni abilitate:

Listato 19: docs/source/conf.py

```
extensions = [  
    'sphinx.ext.duration',  
    'sphinx.ext.doctest',  
    'sphinx.ext.autodoc',  
]
```

Successivamente, si sposta il contenuto della direttiva `.. py:function` nella funzione docstring nel file Python originale, in questo modo:

Listato 20: lumache.py

```
def get_random_ingredients(kind=None):  
    """  
    Return a list of random ingredients as strings.  
  
    :param kind: Optional "kind" of ingredients.  
    :type kind: list[str] or None  
    :raise lumache.InvalidKindError: If the kind is invalid.  
    :return: The ingredients list.  
    :rtype: list[str]  
  
    """  
    return ["shells", "gorgonzola", "parsley"]
```

³⁹ <https://peps.python.org/pep-0257/#what-is-a-docstring>

Infine, si sostituisce la direttiva `.. py:function` dalla documentazione Sphinx con `autofunction`:

Listato 21: docs/source/usage.rst

```
you can use the ``lumache.get_random_ingredients()`` function:
```

```
.. autofunction:: lumache.get_random_ingredients
```

Se ora si crea la documentazione HTML, l'output sarà uguale! Con il vantaggio che viene generato dal codice stesso. Sphinx ha preso il reStructuredText dalla docstring e lo ha incluso, generando anche gli opportuni riferimenti incrociati.

Si può anche generare automaticamente la documentazione da altri oggetti. Per esempio, aggiungiamo il codice per l'eccezione `InvalidKindError`:

Listato 22: lumache.py

```
class InvalidKindError(Exception):
    """Raised if the kind is invalid."""
    pass
```

E si sostituisce la direttiva `.. py:exception` con `autoexception` come segue:

Listato 23: docs/source/usage.rst

```
or ``"veggies"``. Otherwise, :py:func:`lumache.get_random_ingredients`
will raise an exception.
```

```
.. autoexception:: lumache.InvalidKindError
```

E ancora, dopo aver eseguito `make html`, l'output sarà lo stesso di prima.

Generazione di tutti i riferimenti delle API

Sebbene l'utilizzo di `sphinx.ext.autodoc` semplifichi molto mantenere sincronizzati il codice e la documentazione, è comunque necessario scrivere una direttiva `auto*` per ogni oggetto che si desidera documentare. Sphinx fornisce ancora un altro livello di automazione: l'estensione `autosummary`.

La direttiva `autosummary` genera documenti che contengono tutte le direttive `autodoc` necessarie. Per usarla, si abilita prima l'estensione `autosummary`:

Listato 24: docs/source/conf.py

```
extensions = [
    'sphinx.ext.duration',
    'sphinx.ext.doctest',
    'sphinx.ext.autodoc',
```

(continues on next page)

(continua dalla pagina precedente)

```
'sphinx.ext.autosummary',  
]
```

Successivamente, si crea un nuovo file `api.rst` con questi contenuti:

Listato 25: docs/source/api.rst

```
API  
===  
  
.. autosummary::  
   :toctree: generated  
  
lumache
```

Ricordarsi di includere il nuovo documento nel toctree principale:

Listato 26: docs/source/index.rst

```
Contents  
-----  
  
.. toctree::  
  
   usage  
   api
```

Infine, dopo aver creato la documentazione HTML eseguendo `make html`, si avranno due nuove pagine:

- `api.html`, corrispondente a `docs/source/api.rst` e contenente una tabella con gli oggetti inclusi nella direttiva `autosummary` (in questo caso, solo uno).
- `generated/lumache.html`, corrispondente a un file reST appena creato `generated/lumache.rst` e contenente un riepilogo dei membri del modulo, in questo caso una funzione e un'eccezione.

Ciascuno dei link nella pagina di riepilogo porterà nei posti in cui è stata utilizzata originariamente la direttiva `autodoc` corrispondente, in questo caso nel documento `usage.rst`.

Nota: I file generati sono basati sui [template Jinja2](https://jinja.palletsprojects.com/)⁴⁰ che *possono essere personalizzati*, ma questo non rientra nell'ambito di questo tutorial.

⁴⁰ <https://jinja.palletsprojects.com/>

lumache

Lumache - Python library for cooks and food lovers.

FUNCTIONS

<code>get_random_ingredients</code> ([kind])	Return a list of random ingredients as strings.
--	---

EXCEPTIONS

<code>InvalidKindError</code>	Raised if the kind is invalid.
-------------------------------	--------------------------------

Fig. 5: Pagina di riepilogo creata da autosummary

3.7 Appendice: Distribuzione online di un progetto Sphinx

Quando il progetto della documentazione è pronto per essere mostrato al mondo, ci sono molte opzioni disponibili per farlo. Dato che l'HTML generato da Sphinx è statico, si può disaccoppiare il processo di creazione della documentazione HTML dall'hosting di tali file nella piattaforma scelta. Non c'è bisogno di un server sofisticato con Python: praticamente ogni servizio di web hosting sarà sufficiente.

Pertanto, la questione non è tanto come o dove servire l'HTML statico, quanto piuttosto come scegliere un flusso di lavoro che aggiorni automaticamente la documentazione distribuita ogni volta che c'è una modifica nei file sorgenti.

Le sezioni seguenti descrivono alcune soluzioni per distribuire la documentazione in linea e forniscono alcune informazioni di base. Per passare direttamente alla parte pratica, si può saltare a *Pubblicare i sorgenti della documentazione*.

Possibilità di distribuzione compatibili con Sphinx

Esistono diverse opzioni possibili per pubblicare la documentazione Sphinx. Alcune sono:

Read the Docs

[Read the Docs](https://readthedocs.org/)⁴¹ è un servizio online specializzato nell'hosting di documentazione tecnica scritta in Sphinx, nonché in MkDocs. Hanno una serie di funzionalità extra, come documentazione versionata, analisi del traffico e della ricerca, domini personalizzati, reindirizzamenti definiti dall'utente e altro ancora.

GitHub Pages

[GitHub Pages](https://pages.github.com/)⁴² è un semplice web hosting statico strettamente integrato con [GitHub](https://github.com/)⁴³: l'HTML statico viene servito da uno dei branch di un progetto e solitamente i sorgenti

⁴¹ <https://readthedocs.org/>

⁴² <https://pages.github.com/>

⁴³ <https://github.com/>

vengono archiviati in un altro branch in modo che l'output possa essere aggiornato ogni volta che i sorgenti cambiano (ad esempio utilizzando [GitHub Actions](#)⁴⁴). È gratuito e supporta domini personalizzati.

GitLab Pages

[GitLab Pages](#)⁴⁵ è concettualmente simile a GitHub Pages, ma è integrato con [GitLab](#)⁴⁶ e solitamente automatizzato con [GitLab CI](#)⁴⁷.

Netlify

[Netlify](#)⁴⁸ è un sofisticato hosting per siti statici potenziato da tecnologie web lato client come JavaScript (il cosiddetto "[Jamstack](#)"⁴⁹). Offrono supporto per sistemi di content management headless ed elaborazione serverless.

Your own server

Si può sempre utilizzare il proprio server web per ospitare la documentazione HTML di Sphinx. È la soluzione che offre maggiore flessibilità, ma anche maggiore complessità.

Tutte queste opzioni sono a costo zero, con la possibilità di pagare per funzionalità extra.

La filosofia "Docs as Code".

Le offerte gratuite della maggior parte delle soluzioni sopra elencate richiedono che i sorgenti della documentazione siano disponibili pubblicamente. Inoltre, questi servizi richiedono l'uso di un [Version Control System](#)⁵⁰, una tecnologia che tiene traccia dell'evoluzione di una raccolta di file come una serie di istantanee [snapshot] ("commit"). La pratica di scrivere documentazione in file di testo semplice con gli stessi strumenti utilizzati per lo sviluppo del software è comunemente nota come "[Docs as Code](#)"⁵¹.

Il Version Control System più popolare al giorno d'oggi è [Git](#)⁵², uno strumento gratuito e open source che costituisce la spina dorsale di servizi come GitHub e GitLab. Poiché sia Read the Docs che Netlify hanno integrazioni con GitHub e GitLab, e sia GitHub che GitLab hanno un prodotto Pages integrato, il modo più efficace per creare automaticamente la documentazione online è caricare i sorgenti su uno di questi servizi di hosting Git.

⁴⁴ <https://github.com/features/actions>

⁴⁵ <https://about.gitlab.com/stages-devops-lifecycle/pages/>

⁴⁶ <https://gitlab.com/>

⁴⁷ <https://about.gitlab.com/stages-devops-lifecycle/continuous-integration/>

⁴⁸ <https://www.netlify.com/>

⁴⁹ <https://jamstack.org/>

⁵⁰ https://en.wikipedia.org/wiki/Version_control

⁵¹ <https://www.writethedocs.org/guide/docs-as-code/>

⁵² <https://git-scm.com/>

Pubblicare i sorgenti della documentazione

GitHub

Il modo più rapido per caricare un progetto esistente su GitHub è:

1. Registrare un account GitHub⁵³.
2. Creare un nuovo repository⁵⁴.
3. Aprire la pagina "Upload files"⁵⁵ del nuovo repository.
4. Selezionare i file nel browser dei file del sistema operativo (in questo caso `README.rst`, `lumache.py`, i `makefile` nella directory `docs` e tutto in `docs/source`) e trascinarli nell'interfaccia di GitHub per caricarli tutti.
5. Click sul pulsante *Commit changes*.

Nota: La directory `docs/build` non dev'essere caricata, in quanto contiene l'output generato da Sphinx e cambierà ogni volta che si modificano i sorgenti, complicando il flusso di lavoro.

Questi passaggi non richiedono l'accesso alla riga di comando o l'installazione di software aggiuntivo. Per saperne di più, leggere [questo tutorial di avvio rapido](#)⁵⁶ o consultare la [documentazione ufficiale di GitHub](#)⁵⁷

GitLab

Analogamente a GitHub, il modo più veloce per caricare il progetto su GitLab è utilizzare l'interfaccia web:

1. Registrazione per un account GitLab⁵⁸.
2. Creare un nuovo progetto vuoto⁵⁹.
3. Caricare i file di progetto (nel nostro caso `README.rst`, `lumache.py`, i `makefile` nella directory `docs` e tutto il contenuto di `docs/source`) uno alla volta utilizzando il pulsante *Upload File*⁶².

Ancora una volta, questi passaggi non richiedono software aggiuntivo sul computer. Per saperne di più, si può:

⁵³ <https://github.com/signup>

⁵⁴ <https://github.com/new>

⁵⁵ <https://docs.github.com/en/repositories/working-with-files/managing-files/adding-a-file-to-a-repository>

⁵⁶ <https://docs.github.com/en/get-started/quickstart>

⁵⁷ <https://docs.github.com/en/get-started>

⁵⁸ https://gitlab.com/users/sign_up

⁵⁹ <https://gitlab.com/projects/new>

⁶² Al momento in cui scriviamo, il caricamento di intere directory su GitLab utilizzando la sola interfaccia web⁶³ non è ancora implementato.

⁶³ <https://gitlab.com/gitlab-org/gitlab/-/issues/228490>

- Seguire [questo tutorial](#)⁶⁰ per installare Git sul computer.
- Sfogliare la [documentazione Utente di GitLab](#)⁶¹ per comprendere le possibilità della piattaforma.

Nota: La directory `docs/build` non dev'essere caricata, in quanto contiene l'output generato da Sphinx e cambierà ogni volta che si modificano i sorgenti, complicando il flusso di lavoro.

Pubblicare la documentazione HTML

Read the Docs

[Read the Docs](#)⁶⁴ offre l'integrazione sia con GitHub che con GitLab. Il modo più rapido per iniziare è seguire [il tutorial RTD](#)⁶⁵, che è vagamente basato su questo. Si possono pubblicare i propri sorgenti su GitHub come spiegato [nella sezione precedente](#), poi passare direttamente a [Creating a Read the Docs account](#)⁶⁶. Per GitLab, il processo è simile.

GitHub Pages

[GitHub Pages](#)⁶⁷ richiede di [pubblicare i sorgenti](#) su [GitHub](#)⁶⁸. Successivamente, ci sarà bisogno di un processo automatizzato che esegua il passaggio `make html` ogni volta che si modificano i sorgenti. Questo si fa utilizzando [GitHub Actions](#)⁶⁹.

Dopo aver pubblicato i sorgenti su GitHub, si crea un file chiamato `.github/workflows/sphinx.yml` nel repository col seguente contenuto:

Listato 27: `.github/workflows/`

```
name: "Sphinx: Render docs"

on: push

jobs:
  build:
    runs-on: ubuntu-latest
    permissions:
      contents: write
    steps:
```

(continues on next page)

⁶⁰ <https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html>

⁶¹ <https://docs.gitlab.com/ee/user/index.html>

⁶⁴ <https://readthedocs.org/>

⁶⁵ <https://docs.readthedocs.io/en/stable/tutorial/index.html>

⁶⁶ <https://docs.readthedocs.io/en/stable/tutorial/index.html#creating-a-read-the-docs-account>

⁶⁷ <https://pages.github.com/>

⁶⁸ <https://github.com/>

⁶⁹ <https://github.com/features/actions>

(continua dalla pagina precedente)

```
- uses: actions/checkout@v4
- name: Build HTML
  uses: ammaraskar/sphinx-action@master
- name: Upload artifacts
  uses: actions/upload-artifact@v4
  with:
    name: html-docs
    path: docs/build/html/
- name: Deploy
  uses: peaceiris/actions-gh-pages@v3
  if: github.ref == 'refs/heads/main'
  with:
    github_token: ${ secrets.GITHUB_TOKEN }
    publish_dir: docs/build/html
```

Contiene un flusso di lavoro [workflow] GitHub Actions con un singolo job di quattro passaggi:

1. Checkout del codice.
2. Build della documentazione HTML con Sphinx.
3. Allega l'output HTML degli artefatti al job GitHub Actions, per un'ispezione più semplice.
4. Se la modifica avviene nel branch di default, si prendono i contenuti di docs/build/html e si inviano [push] nel branch gh-pages.

Successivamente, è necessario specificare le dipendenze affinché il passaggio con `make html` abbia successo. Per questo, si crea un file `docs/requirements.txt` e si aggiunge il seguente contenuto:

Listato 28: docs/requirements.txt

```
furo==2021.11.16
```

E finalmente si è pronti per [abilitare GitHub Pages sul repository](#)⁷⁰. Per farlo, si va su *Settings*, poi *Pages* sulla barra laterale di sinistra, si seleziona il branch `gh-pages` nel menu a discesa "Source", e click su *Save*. Dopo alcuni minuti, si dovrebbe essere in grado di vedere il codice HTML all'URL designato.

GitLab Pages

[GitHub Pages](#)⁷¹, d'altra parte, richiede di *pubblicare i sorgenti* su [GitHub](#)⁷². Quando si è pronti, si può automatizzare il processo di esecuzione di `make html` usando [GitLab CI](#)⁷³.

Dopo la pubblicazione dei sorgenti su GitLab, si crea un file chiamato `.gitlab-ci.yml` nel repository con questi contenuti:

Listato 29: `.gitlab-ci.yml`

```
stages:
  - deploy

pages:
  stage: deploy
  image: python:3.9-slim
  before_script:
    - apt-get update && apt-get install make --no-install-recommends -y
    - python -m pip install sphinx furo
  script:
    - cd docs && make html
  after_script:
    - mv docs/build/html/ ./public/
  artifacts:
    paths:
      - public
  rules:
    - if: $CI_COMMIT_REF_NAME == $CI_DEFAULT_BRANCH
```

Contiene un flusso di lavoro GitLab CI con un lavoro composto da diversi passaggi:

1. Installazione delle dipendenze necessarie.
2. Build della documentazione HTML con Sphinx.
3. Sposta l'output in una posizione nota degli artefatti.

⁷⁰ <https://docs.github.com/en/pages/getting-started-with-github-pages/configuring-a-publishing-source-for-your-github-pages-s>

⁷¹ <https://pages.github.com/>

⁷² <https://github.com/>

⁷³ <https://about.gitlab.com/stages-devops-lifecycle/continuous-integration/>

Nota: Si dovrà [validare l'account](#)⁷⁴ inserendo un metodo di pagamento (verrà addebitato un piccolo importo che verrà poi rimborsato).

Successivamente, se la pipeline ha esito positivo, si dovrebbe essere in grado di vedere l'HTML all'URL designato.

3.8 Come proseguire

Questo tutorial ha coperto i primissimi passi per creare un progetto di documentazione con Sphinx. Per continuare, e saperne di più su Sphinx, consultare il [resto della documentazione](#).

⁷⁴ <https://about.gitlab.com/blog/2021/05/17/prevent-crypto-mining-abuse/#validating-an-account>

CAPITOLO 4

Uso di Sphinx

Questa guida serve a mostrare come iniziare con Sphinx e copre tutto, dall'installazione di Sphinx e la configurazione del primo progetto Sphinx all'utilizzo di alcune delle funzionalità avanzate che Sphinx fornisce nativamente. Se si cercano indicazioni su come estendere Sphinx, si faccia riferimento a *Estendere Sphinx*.

4.1 reStructuredText

reStructuredText (reST) è il linguaggio di markup di semplice testo utilizzato di default sia da Docutils che da Sphinx. Docutils fornisce la sintassi di base di reStructuredText, mentre Sphinx la estende per supportare funzionalità aggiuntive.

Le guide seguenti esaminano gli aspetti più importanti di reST. Per il riferimento ufficiale a reStructuredText, fare riferimento alla [docutils documentation](https://docutils.sourceforge.io/rst.html)⁷⁵.

reStructuredText Primer

reStructuredText è il linguaggio di markup di semplice testo utilizzato di default da Sphinx. Questa sezione è una breve introduzione ai concetti e alla sintassi di reStructuredText (reST), per fornire agli autori informazioni sufficienti per creare documenti in modo produttivo. Poiché reST è stato progettato per essere un linguaggio di markup semplice e discreto, questo non richiederà molto tempo.

⁷⁵ <https://docutils.sourceforge.io/rst.html>

Vedi anche:

L'autorevole [Documentazione Utente di reStructuredText](#)⁷⁶. I link "ref" in questo documento si collegano alla descrizione dei singoli costrutti nel reference di reST.

Paragrafi

Il paragrafo ([ref](#)⁷⁷) è il blocco più elementare in un documento reST. I paragrafi sono semplicemente porzioni di testo separate da una o più righe vuote. Come in Python, l'indentazione è significativa in reST, quindi tutte le righe dello stesso paragrafo devono essere allineate a sinistra allo stesso livello di rientro.

Markup in linea

Il markup in linea standard di reST è abbastanza semplice: usa

- un asterisco: `*testo*` per enfasi (corsivo),
- due asterischi: `**testo**` per enfasi forte (grassetto), e
- virgolette inverse [backquote]: ``testo`` per esempi di codice.

Se nel testo corrente compaiono asterischi o virgolette inverse che potrebbero essere confuse con i delimitatori di markup in linea, devono essere preceduti da una barra rovesciata [backslash].

Da tener presente alcune restrizioni di questo markup:

- non può essere nidificato,
- il contenuto non deve iniziare o terminare con spazi bianchi: `* testo*` è sbagliato,
- deve essere separato dal testo circostante da caratteri non verbali [non-word]. Utilizzando uno spazio preceduto da una backslash si può aggirare il problema: `questa\ è\ *una*\ parola`.

Tali restrizioni potrebbero essere rimosse nelle future versioni di docutils.

È anche possibile sostituire o espandere parte di questo markup in linea con i ruoli. Fare riferimento a [Ruoli](#) per ulteriori informazioni.

Elenchi e blocchi evidenziati [quoted]

Il markup per gli elenchi ([ref](#)⁷⁸) è banale: basta posizionare un asterisco all'inizio di un paragrafo e indentare correttamente. Lo stesso vale per gli elenchi numerati; possono anche essere numerati automaticamente utilizzando il segno #:

⁷⁶ <https://docutils.sourceforge.io/rst.html>

⁷⁷ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#paragraphs>

⁷⁸ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#bullet-lists>

```
* This is a bulleted list.
* It has two items, the second
  item uses two lines.

1. This is a numbered list.
2. It has two items too.

#. This is a numbered list.
#. It has two items too.
```

Si possono avere degli elenchi nidificati, ma si tenga presente che devono essere separati dagli elementi dell'elenco principale da righe vuote:

```
* this is
* a list

    * with a nested list
    * and some subitems

* and here the parent list continues
```

Gli elenchi di definizioni ([ref⁷⁹](#)) si creano in questo modo:

```
term (up to a line of text)
    Definition of the term, which must be indented

    and can even consist of multiple paragraphs

next term
    Description.
```

Da notare che il termine non può contenere più di una riga di testo.

I paragrafi evidenziati [quoted paragraphs] ([ref⁸⁰](#)) si creano semplicemente facendoli rientrare più dei paragrafi circostanti.

I blocchi di righe [Line blocks] ([ref⁸¹](#)) sono un modo per preservare le interruzioni di riga:

```
| These lines are
| broken exactly like in
| the source file.
```

Sono disponibili anche molti altri blocchi speciali:

- elenchi di campi ([ref⁸²](#), con avvertenze [caveats] annotate in *Liste di campi*)

⁷⁹ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#definition-lists>

⁸⁰ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#block-quotes>

⁸¹ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#line-blocks>

⁸² <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#field-lists>

- elenchi di opzioni ([ref⁸³](#))
- blocchi letterali evidenziati [quoted] ([ref⁸⁴](#))
- blocchi doctest ([ref⁸⁵](#))

Blocchi letterali

I blocchi di codice letterale ([ref⁸⁶](#)) vengono introdotti terminando un paragrafo con il marcatore speciale `::`. Il blocco letterale deve essere rientrato (e, come tutti i paragrafi, separato da quelli circostanti da righe vuote):

```
This is a normal text paragraph. The next paragraph is a code sample::  
  
    It is not processed in any way, except  
    that the indentation is removed.  
  
    It can span multiple lines.  
  
This is a normal text paragraph again.
```

La gestione del marcatore `::` è intelligente:

- Se si presenta come un paragrafo a sé stante, quel paragrafo viene completamente escluso dal documento.
- Se è preceduto da uno spazio, il marcatore viene rimosso.
- Se è preceduto da spazi non "bianchi", il marcatore viene sostituito dai "due punti".

In questo modo, la seconda frase del primo paragrafo dell'esempio precedente verrebbe resa come "Il paragrafo successivo è un esempio di codice:".

L'evidenziazione del codice può essere abilitata per questi blocchi letterali a livello di documento utilizzando la direttiva `highlight` e a livello di progetto utilizzando l'opzione di configurazione `highlight_language`. La direttiva `code-block` è utilizzabile per impostare l'evidenziazione blocco per blocco. Queste direttive verranno trattate più avanti.

⁸³ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#option-lists>

⁸⁴ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#quoted-literal-blocks>

⁸⁵ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#doctest-blocks>

⁸⁶ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#literal-blocks>

Blocchi doctest

I blocchi doctest ([ref⁸⁷](#)) sono sessioni Python interattive tagliate e incollate in docstring. Non richiedono la sintassi dei *blocchi letterali*. Il blocco doctest deve terminare con una riga vuota e *non* deve terminare con un prompt inutilizzato:

```
>>> 1 + 1
2
```

Tabelle

Per le *griglie delle tabelle* ([ref⁸⁸](#)), si deve "disegnare" manualmente la griglia della cella. Appaiono in questo modo:

```
+-----+-----+-----+-----+
| Header row, column 1 | Header 2 | Header 3 | Header 4 |
| (header rows optional) |          |          |          |
+=====+=====+=====+=====+
| body row 1, column 1 | column 2 | column 3 | column 4 |
+-----+-----+-----+-----+
| body row 2          | ...     | ...     |          |
+-----+-----+-----+-----+
```

Le *tabelle semplici* ([ref⁸⁹](#)) sono più facili da scrivere, ma limitate: devono contenere più di una riga e le celle della prima colonna non possono contenere più righe. Appaiono in questo modo:

```
=====
A      B      A and B
=====
False  False  False
True   False  False
False  True   False
True   True   True
=====
```

Sono supportate altre due sintassi: *tabelle CSV* e *Tabelle elenco*. Utilizzano un *blocco di markup esplicito*. Fare riferimento a [Tabelle](#) per ulteriori informazioni.

⁸⁷ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#doctest-blocks>

⁸⁸ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#grid-tables>

⁸⁹ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#simple-tables>

Hyperlink

Link esterni

Per i link web in linea, si usa ``Testo del link <https://domain.invalid/>`_`. Se il testo del link è un indirizzo web, non è necessario alcun markup speciale, il parser trova link e indirizzi e-mail nel testo normale.

Importante: Deve esserci uno spazio tra il testo del link e l'apertura `<` dell'URL.

Si possono anche separare il link e la definizione del target ([ref⁹⁰](#)), in questo modo:

```
This is a paragraph that contains `a link`_.
```

```
.. _a link: https://domain.invalid/
```

Link Interni

Il link interno viene fatto tramite uno speciale ruolo reST fornito da Sphinx, vedere la sezione sul markup specifico, *Riferimenti incrociati di posizioni arbitrarie*.

Sezioni

Le intestazioni delle sezioni ([ref⁹¹](#)) vengono create sottolineando (e facoltativamente sovrallineando) il titolo della sezione con un carattere di punteggiatura, lungo almeno quanto il testo:

```
=====
This is a heading
=====
```

Normalmente non ci sono livelli di intestazione assegnati a determinati caratteri poiché la struttura è determinata dalla loro successione. Tuttavia, questa convenzione viene utilizzata nella [Guida alla documentazione degli sviluppatori Python⁹²](#) che si può seguire:

- # con soprallineamento, per le parti
- * con soprallineamento, per i capitoli
- = per le sezioni
- - per le sottosezioni
- ^ per le sotto-sottosezioni

⁹⁰ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#hyperlink-targets>

⁹¹ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#sections>

⁹² <https://devguide.python.org/documentation/markup/#sections>

- " per i paragrafi

Naturalmente, si è liberi di utilizzare i propri caratteri marcatori (consultare la documentazione reST) e di utilizzare un livello di nidificazione più profondo, ma si tenga presente che la maggior parte dei formati target (HTML, LaTeX) hanno una profondità di nidificazione limitata.

Liste di campi

Le liste di campi ([ref⁹³](#)) sono sequenze di campi contrassegnati in questo modo:

```
:fieldname: Field content
```

Sono comunemente usati nella documentazione Python:

```
def my_function(my_arg, my_other_arg):
    """A function just for me.

    :param my_arg: The first of my arguments.
    :param my_other_arg: The second of my arguments.

    :returns: A message (just for me, of course).
    """
```

Sphinx estende il comportamento standard dei documenti e intercetta gli elenchi di campi specificati all'inizio dei documenti. Fare riferimento a *Liste di campi* per ulteriori informazioni.

Ruoli

Un ruolo o "ruolo di testo interpretato personalizzato" ([ref⁹⁴](#)) è un pezzo in linea di markup esplicito. Ciò significa che il testo racchiuso deve essere interpretato in un modo specifico. Sphinx lo usa per fornire markup semantici e riferimenti incrociati di identificatori, come descritto nella sezione appropriata. La sintassi generale è `:rolename: `content``.

Docutils supporta i seguenti ruoli:

- `emphasis95` -- equivalente a `*emphasis*`
- `strong96` -- equivalente a `**strong**`
- `literal97` -- equivalente a ``literal``
- `subscript98` -- testo in pedice
- `superscript99` -- testo in apice

⁹³ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#field-lists>

⁹⁴ <https://docutils.sourceforge.io/docs/ref/rst/roles.html>

⁹⁵ <https://docutils.sourceforge.io/docs/ref/rst/roles.html#emphasis>

⁹⁶ <https://docutils.sourceforge.io/docs/ref/rst/roles.html#strong>

⁹⁷ <https://docutils.sourceforge.io/docs/ref/rst/roles.html#literal>

⁹⁸ <https://docutils.sourceforge.io/docs/ref/rst/roles.html#subscript>

⁹⁹ <https://docutils.sourceforge.io/docs/ref/rst/roles.html#superscript>

- `title-reference`¹⁰⁰ -- per titoli di libri, periodici e altri materiali

Fare riferimento a *Ruoli* per i ruoli aggiunti da Sphinx.

Markup Esplicito

Il "markup esplicito" (`ref`¹⁰¹) viene utilizzato in reST per la maggior parte dei costrutti che richiedono una gestione speciale, come note a piè di pagina, paragrafi appositamente evidenziati, commenti e direttive generiche.

Un blocco di markup esplicito inizia con una riga con `..` seguita da spazi bianchi e termina con il paragrafo successivo allo stesso livello di indentazione. (È necessario che ci sia una riga vuota tra il markup esplicito e i paragrafi normali. Tutto questo può sembrare un po' complicato, ma è abbastanza intuitivo quando lo si scrive).

Direttive

Una direttiva (`ref`¹⁰²) è un blocco generico di markup esplicito. Insieme ai ruoli, è uno dei meccanismi di estensione di reST e Sphinx ne fa un uso massiccio.

Docutils supporta le seguenti direttive:

- Avvertenze: `attention`¹⁰³, `caution`¹⁰⁴, `danger`¹⁰⁵, `error`¹⁰⁶, `hint`¹⁰⁷, `important`¹⁰⁸, `note`¹⁰⁹, `tip`¹¹⁰, `warning`¹¹¹ e il generico `admonition`¹¹². (La maggior parte dei temi di stile presenta in particolare solo "note" e "warning").
- Immagini:
 - `image`¹¹³ (vedere anche *Immagini* di seguito)
 - `figure`¹¹⁴ (un'immagine con didascalia e una legenda opzionale)
- Elementi aggiuntivi del corpo del testo:
 - `contents`¹¹⁵ (un sommario locale, cioè solo per il file corrente)

¹⁰⁰ <https://docutils.sourceforge.io/docs/ref/rst/roles.html#title-reference>

¹⁰¹ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#explicit-markup-blocks>

¹⁰² <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#directives>

¹⁰³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#attention>

¹⁰⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#caution>

¹⁰⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#danger>

¹⁰⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#error>

¹⁰⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#hint>

¹⁰⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#important>

¹⁰⁹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#note>

¹¹⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#tip>

¹¹¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#warning>

¹¹² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#admonitions>

¹¹³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#image>

¹¹⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#figure>

¹¹⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

- `container`¹¹⁶ (un contenitore con una classe custom, utile per generare un `<div>` esterno in HTML)
- `rubric`¹¹⁷ (un'intestazione senza relazione con la suddivisione del documento)
- `topic`¹¹⁸, `sidebar`¹¹⁹ (elementi speciali del corpo del testo in evidenza)
- `parsed-literal`¹²⁰ (blocco letterale che supporta il markup in linea)
- `epigraph`¹²¹ (un blocco evidenziato con riga di attribuzione opzionale)
- `highlights`¹²², `pull-quote`¹²³ (blocchi evidenziati con il proprio attributo di classe)
- `compound`¹²⁴ (un paragrafo composto)
- Tabelle speciali:
 - `table`¹²⁵ (una tabella con un titolo)
 - `csv-table`¹²⁶ (una tabella generata da valori separati da virgole)
 - `list-table`¹²⁷ (una tabella generata da un elenco di elenchi)
- Direttive speciali:
 - `raw`¹²⁸ (include markup del formato target non elaborato)
 - `include`¹²⁹ (include reStructuredText da un altro file) -- in Sphinx, quando viene fornito un path assoluto del file incluso, questa direttiva lo considera relativo alla directory sorgente
 - `class`¹³⁰ (assegna un attributo di classe all'elemento successivo)

Nota: Quando il dominio di default contiene una direttiva `class`, questa direttiva verrà ombreggiata. Pertanto, Sphinx lo riesporta come `rst-class`.

- Specifiche per l'HTML:
 - `meta`¹³¹ (generazione di tag HTML `<meta>`, consultare anche *Metadati HTML* di seguito)

¹¹⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#container>

¹¹⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#rubric>

¹¹⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

¹¹⁹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#sidebar>

¹²⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#parsed-literal>

¹²¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#epigraph>

¹²² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#highlights>

¹²³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#pull-quote>

¹²⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#compound-paragraph>

¹²⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table>

¹²⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#csv-table>

¹²⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#list-table>

¹²⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#raw-data-pass-through>

¹²⁹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

¹³⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#class>

¹³¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#meta>

- `title`¹³² (sostituisce il titolo del documento)
- Markup modificatori:
 - `default-role`¹³³ (imposta un nuovo ruolo di default)
 - `role`¹³⁴ (crea un nuovo ruolo)

Dato che questi sono solo per file, è meglio utilizzare le funzionalità di Sphinx per impostare `default_role`.

Avvertimento: Non utilizzare le direttive `sectnum`¹³⁵, `header`¹³⁶ e `footer`¹³⁷.

Le direttive aggiunte da Sphinx sono descritte in *Direttive*.

Fondamentalmente, una direttiva è composta da un nome, argomenti, opzioni e contenuto. (Tene presente questa terminologia, verrà utilizzata nel prossimo capitolo che descrive le direttive custom). Analizziamo questo esempio,

```
.. function:: foo(x)
           foo(y, z)
:module: some.module.name
```

Return a line of text input from the user.

`function` è il nome della direttiva. Qui vengono forniti due argomenti, il resto della prima e della seconda riga, nonché un'opzione `module` (come si vede, le opzioni vengono fornite nelle righe immediatamente successive agli argomenti e indicate dai due punti). Le opzioni devono essere indentate allo stesso livello del contenuto della direttiva.

Il contenuto della direttiva segue dopo una riga vuota ed è rientrato rispetto all'inizio della direttiva o, se sono presenti opzioni, della stessa quantità delle opzioni.

Attenzione perché il rientro non è un numero fisso di spazi bianchi, ad es. tre, ma qualsiasi numero di spazi bianchi. Ciò può sorprendere quando viene utilizzato un rientro fisso in tutto il documento e può fare la differenza per le direttive sensibili agli spazi bianchi. Confrontare:

```
.. code-block::
:caption: A cool example
```

The output of this line starts with four spaces.

```
.. code-block::
```

The output of this line has no spaces at the beginning.

¹³² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#metadata-document-title>

¹³³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#default-role>

¹³⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#role>

¹³⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#sectnum>

¹³⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#header>

¹³⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#footer>

Nel primo blocco di codice il rientro del contenuto è stato fissato a tre spazi dalla riga delle opzioni, di conseguenza il contenuto inizia con quattro spazi. In quest'ultimo esempio il rientro è stato fissato dal contenuto stesso a sette spazi, quindi non inizia con uno spazio.

Immagini

reST supporta una direttiva per l'immagine ([ref¹³⁸](#)), utilizzata in questo modo:

```
.. image:: gnu.png
   (options)
```

Quando è utilizzato all'interno di Sphinx, il nome del file (qui `gnu.png`) deve essere relativo al file sorgente o assoluto, il che significa che sono relativi alla directory sorgente principale. Ad esempio, il file `sketch/spam.rst` potrebbe fare riferimento all'immagine `images/spam.png` come a `../images/spam.png` o a `/images/spam.png`.

Sphinx copierà automaticamente i file immagine in una sottodirectory della directory di output durante la creazione (ad esempio la directory `_static` per l'output HTML).

L'interpretazione delle opzioni per le dimensioni dell'immagine (larghezza e altezza) è la seguente: se la dimensione non ha unità o l'unità è pixel, la dimensione specificata verrà rispettata solo per i canali di output che supportano i pixel. Altre unità (come pt per punti) verranno utilizzate per l'output HTML e LaTeX (quest'ultimo sostituisce pt con bp poiché questa è l'unità TeX tale che 72bp=1in).

Sphinx estende il comportamento standard di docutils consentendo un asterisco per l'estensione:

```
.. image:: gnu.*
```

Sphinx cerca poi tutte le immagini che corrispondono al pattern fornito e ne determina il tipo. Ogni builder sceglie quindi l'immagine migliore tra quelle candidate. Ad esempio, se è stato fornito il nome del file `gnu.*` ed esistono due file `gnu.pdf` e `gnu.png` nella struttura dei sorgenti, il builder LaTeX sceglierebbe il primo, mentre quello HTML preferirebbe il secondo. I tipi di immagine supportati e la scelta della priorità sono definiti in *[I Builder](#)*.

Da notare che i nomi dei file di immagini non devono contenere spazi.

Cambiato nella versione 0.4: Aggiunto il supporto per i nomi di file che terminano con un asterisco.

Cambiato nella versione 0.6: I path delle immagini ora possono essere assoluti.

Cambiato nella versione 1.5: il target latex 1.5 supporta i pixel (il valore di default è 96px=1in).

¹³⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#image>

Note a piè di pagina

Per le note a piè di pagina ([ref¹³⁹](#)), si usa `[#name]_` per contrassegnare la posizione della nota a piè di pagina e si aggiunge il corpo della nota in fondo al documento dopo un'intestazione della rubrica "Note a piè di pagina", in questo modo:

```

Lorem ipsum [#f1]_ dolor sit amet ... [#f2]_

.. rubric:: Footnotes

.. [#f1] Text of the first footnote.
.. [#f2] Text of the second footnote.
```

Le note a piè di pagina si possono anche numerare esplicitamente (`[1]_`) o utilizzare note a piè di pagina numerate automaticamente senza nomi (`[#]_`).

Citazioni

Sono supportate le citazioni reST standard ([ref¹⁴⁰](#)), con la caratteristica aggiuntiva che sono "globali", cioè è possibile fare riferimento a tutte le citazioni da tutti i file. Si usano in questo modo:

```

Lorem ipsum [Ref]_ dolor sit amet.

.. [Ref] Book or article reference, URL or whatever.
```

L'utilizzo delle citazioni è simile a quello delle note a piè di pagina, ma con un'etichetta che non è numerica né inizia con #.

Sostituzioni

reST supporta le "sostituzioni" ([ref¹⁴¹](#)), che sono parti di testo e/o markup a cui si fa riferimento nel testo con `|name|`. Sono definite come note a piè di pagina con blocchi di markup espliciti, in questo modo:

```
.. |name| replace:: replacement *text*
```

o questo:

```
.. |caution| image:: warning.png
   :alt: Warning!
```

Consultare il riferimento [reST per le sostituzioni¹⁴²](#) per i dettagli.

¹³⁹ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#footnotes>

¹⁴⁰ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#citations>

¹⁴¹ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#substitution-definitions>

¹⁴² <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#substitution-definitions>

Se si vogliono utilizzare delle sostituzioni per tutti i documenti, si inseriscono in *rst_prolog* o in *rst_epilog* oppure si inseriscono in un file separato e lo si include in tutti i documenti interessati, con la direttiva `include`¹⁴³. (Assicuratevi di dare al file incluso un'estensione del nome file diversa da quella degli altri file sorgenti, per evitare che Sphinx lo trovi come documento a sé).

Sphinx definisce alcune sostituzioni di default, vedere *Sostituzioni*.

Commenti

Ogni blocco di markup esplicito che non sia un costrutto di markup valido (come le note a piè di pagina sopra) è considerato un commento (*ref*¹⁴⁴). Per esempio:

```
.. This is a comment.
```

Si può rientrare il testo dopo l'inizio di un commento per formare commenti su più righe:

```
..
    This whole indented block
    is a comment.

    Still in the comment.
```

Metadati HTML

La direttiva `meta`¹⁴⁵ consente di specificare il `metadata element`¹⁴⁶ HTML di una pagina di documentazione Sphinx. Ad esempio, la direttiva:

```
.. meta::
   :description: The Sphinx documentation builder
   :keywords: Sphinx, documentation, builder
```

genererà il seguente output HTML:

```
<meta name="description" content="The Sphinx documentation builder">
<meta name="keywords" content="Sphinx, documentation, builder">
```

Inoltre, Sphinx aggiungerà le parole chiave specificate nella meta direttiva nell'indice di ricerca. In tal modo viene considerato l'attributo `lang` dell'elemento meta. Ad esempio, la direttiva:

```
.. meta::
   :keywords: backup
   :keywords lang=en: pleasefindthiskey pleasefindthiskeytoo
   :keywords lang=de: bittediesenkeyfinden
```

¹⁴³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

¹⁴⁴ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#comments>

¹⁴⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#meta>

¹⁴⁶ <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta>

aggiunge le seguenti parole agli indici di ricerca delle build con configurazioni di lingua diverse:

- `pleasefindthiskey`, `pleasefindthiskeytoo` alle build in *Inglese*;
- `perfavoretrovaquestachiave` alle build in *Italiano*;
- `backup` per build in qualsiasi lingua.

Codifica dei sorgenti

Poiché il modo più semplice per includere caratteri speciali come i trattini em o i segni di copyright in reST è scriverli direttamente come caratteri Unicode, è necessario specificare una codifica. Sphinx presuppone che i sorgenti siano codificati in UTF-8 per default; lo si può cambiare con il valore di configurazione `source_encoding` config value.

Trabocchetti

Ci sono alcuni problemi che si incontrano comunemente durante la creazione di documenti reST:

- **Separazione del markup in linea:** Come detto sopra, l'estensione dei markup in linea devono essere separate dal testo circostante da caratteri non verbali, per aggirare questo problema è necessario utilizzare uno spazio preceduto da backslash. Consultare [the reference](#)¹⁴⁷ per i dettagli.
- **Nessun markup in linea nidificato:** Qualcosa come `*see :func:`foo`*` non è possibile.

Ruoli

Sphinx utilizza ruoli [role] di testo interpretati per inserire markup semantico nei documenti. Sono scritti come `:rolename: `content``.

Nota: Il ruolo di default (``content``) non ha alcun significato speciale di default. Lo si può usare per qualsiasi cosa, ad es. nomi di variabili; si usa il valore di configurazione `default_role` per impostarlo su un ruolo noto -- il ruolo `any` per trovare qualcosa o il ruolo `py:obj` per trovare oggetti Python è molto utile a questo scopo.

Consultare [Domini](#) per i ruoli aggiunti dai domini.

¹⁴⁷ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#substitution-definitions>

Sintassi dei riferimenti incrociati

Vedere *Sintassi dei riferimenti incrociati*.

Tra i ruoli dei riferimenti incrociati ci sono:

- *any*
- *doc*
- *download*
- *envvar*
- *keyword*
- *numref*
- *option* (e il deprecato *cmdoption*)
- *ref*
- *term*
- *token*

Evidenziazione del codice in linea

:code:

Un esempio di codice *inline*. Se utilizzato direttamente, questo ruolo visualizza semplicemente il testo *senza* evidenziazione della sintassi, come valore letterale.

By default, inline code such as `:code:`1 + 2`` just displays without highlighting.

Risultato: Per default, il codice in linea come `1 + 2` viene visualizzato senza evidenziazione.

A differenza della direttiva *code-block*, questo ruolo non rispetta il linguaggio di default impostato dalla direttiva *highlight*.

Per abilitare l'evidenziazione della sintassi, si deve prima utilizzare la direttiva Docutils *role*¹⁴⁸ per definire un ruolo personalizzato associato a un linguaggio specifico:

```
.. role:: python(code)
   :language: python
```

In Python, `:python:`1 + 2`` is equal to `:python:`3``.

Per visualizzare un esempio di codice su più righe, utilizzare invece la direttiva *code-block*.

¹⁴⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#role>

Formule matematiche

:math:

Ruolo per la matematica in linea. Si usa in questo modo:

```
Since Pythagoras, we know that :math:`a^2 + b^2 = c^2`.
```

Risultato: Dai tempi di Pitagora sappiamo che $a^2 + b^2 = c^2$.

:eq:

Uguale a *math:numref*.

Altri markup semantici

I seguenti ruoli non fanno nulla di speciale tranne la formattazione del testo in uno stile diverso:

:abbr:

Un'abbreviazione. Se il contenuto del ruolo contiene una spiegazione tra parentesi, verrà trattato in modo speciale: verrà mostrato in un tool-tip in HTML e verrà visualizzato una sola volta in LaTeX.

Per esempio: `:abbr:`LIFO (last-in, first-out)`` mostra LIFO (last-in, first-out).

Added in version 0.6.

:command:

Il nome di un comando a livello di sistema operativo, ad esempio `rm`.

Per esempio: `rm`

:dfn:

Segna l'istanza che definisce un termine nel testo. (Non viene generata alcuna voce di indice).

Per esempio: *binary mode*

:file:

Il nome di un file o di una directory. All'interno dei contenuti è possibile utilizzare le parentesi graffe per indicare una parte "variabile", ad esempio:

```
... is installed in :file:`/usr/lib/python3.{x}/site-packages` ...
```

Risultato: ... è installato in `/usr/lib/python3.x/site-packages` ...

Nella documentazione finale, la `x` verrà visualizzata in modo diverso per indicare che deve essere sostituita dalla versione di Python.

:guilabel:

Le etichette presentate come parte di un'interfaccia utente interattiva dovrebbero essere contrassegnate utilizzando `guilabel`. Ciò include etichette da interfacce basate su testo

come quelle create utilizzando [curses](https://docs.python.org/3/library/curses.html#module-curses)¹⁴⁹ o altre librerie basate su testo. Qualsiasi etichetta utilizzata nell'interfaccia dovrebbe essere contrassegnata con questo ruolo, comprese le etichette dei pulsanti, i titoli delle finestre, i nomi dei campi, i nomi dei menù e delle selezioni di menù e persino i valori negli elenchi.

Cambiato nella versione 1.0: Un tasto acceleratore per l'etichetta della GUI può essere incluso utilizzando una *e commerciale*; questo verrà rimosso e visualizzato come sottolineatura nell'output (ad esempio: `:guilabel: `&Cancel`` appare *Cancel*). Per includere una *e commerciale* letterale, la si raddoppia.

:kbd:

Segna una sequenza di tasti. La forma assunta dalla sequenza di tasti può dipendere da convenzioni specifiche della piattaforma o dell'applicazione. Quando non esistono convenzioni rilevanti, i nomi dei tasti modificatori dovrebbero essere precisati, per migliorare l'accessibilità per i nuovi utenti e per i non madrelingua. Per esempio, una sequenza di tasti *xemacs* può essere contrassegnata come `:kbd: `C-x C-f``, ma senza riferimento a un'applicazione o piattaforma specifica, la stessa sequenza dovrebbe essere contrassegnata come `:kbd: `Control-x Control-f``, visualizzando rispettivamente *C-x C-f* e *Control-x Control-f*.

:mailheader:

Il nome di un'intestazione di mail in stile RFC 822. Questo markup non implica che l'intestazione venga utilizzata in un messaggio di posta elettronica, ma può essere utilizzata per fare riferimento a qualsiasi intestazione dello stesso "stile". Viene utilizzato anche per le intestazioni definite dalle varie specifiche MIME. Il nome dell'intestazione deve essere inserito nello stesso modo in cui si troverebbe normalmente nella pratica, preferendo la convenzione "camel-casing" laddove esiste più di un utilizzo comune. Per esempio: `:mailheader: `Content-Type`` appare *Content-Type*.

:makevar:

Il nome di una variabile **make**.

Per esempio: **help**

:manpage:

Un riferimento a una pagina di manuale Unix che include la sezione, ad es. `:manpage: `ls(1)`` mostra *ls(1)*. Crea un link ipertestuale a un sito esterno che visualizza la manpage se è definito [*manpages_url*](#).

Cambiato nella versione 7.3: Consentire di specificare un target con `<>`, come link ipertestuali. Per esempio, `:manpage: `blah <ls(1)>`` mostra *blah*.

:menuselection:

Le selezioni dei menù devono essere contrassegnate utilizzando il ruolo *menuselection*. Viene utilizzato per contrassegnare una sequenza completa di selezioni di menù, inclusa la selezione di sotto-menù e la scelta di un'operazione specifica, o qualsiasi sequenza successiva. I nomi delle singole selezioni devono essere separati da `-->`.

Ad esempio, per contrassegnare la selezione "Start > Programs", si usa questo markup:

¹⁴⁹ <https://docs.python.org/3/library/curses.html#module-curses>

```
:menuselection:`Start --> Programs`
```

Risultato: *Start* → *Programs*

Quando c'è una selezione che include qualche indicatore finale, come i puntini di sospensione utilizzati da alcuni sistemi operativi per indicare che il comando apre una finestra di dialogo, l'indicatore deve essere omesso dal nome della selezione.

`menuselection` supporta anche gli acceleratori con la "e commerciale" proprio come *guilabel*.

:mimetype:

Il nome di un tipo MIME o un componente di un tipo MIME (la parte maggiore o minore, presa da sola).

Per esempio: *text/plain*

:newsgroup:

Il nome di un newsgroup Usenet.

Per esempio: *comp.lang.python*

Da fare: Questo non fa parte del dominio standard?

:program:

Il nome di un programma eseguibile. Questo potrebbe differire dal nome file dell'eseguibile per alcune piattaforme. In particolare, per i programmi Windows dovrebbe essere omessa l'estensione *.exe* (o altre).

Per esempio: **curl**

:regexp:

Un'espressione regolare. Le virgolette non dovrebbero essere incluse.

Per esempio: *([abc])+*

:samp:

Un pezzo di testo letterale, come codice. All'interno dei contenuti, si possono utilizzare le parentesi graffe per indicare una parte "variabile", come in *file*. Per esempio, in `:samp:`print(1+{variable})``, la parte *variable* verrebbe enfaticizzata: `print(1+variable)`

Se non c'è bisogno dell'indicazione "parte variabile", si usa invece il ruolo standard *code*.

Cambiato nella versione 1.8: È consentito l'escape delle parentesi graffe con un doppio backslash [*doppia barra rovesciata*]. Per esempio, in `:samp:`print(f"answer=\\{1+{variable}*2\\}")``, la parte *variable* verrebbe enfaticizzata e le parentesi graffe con escape verrebbero visualizzate: `print(f"answer={1+variable*2}")`

Esiste anche un ruolo *index* per generare voci di indice.

I seguenti ruoli generano collegamenti esterni:

:pep:

Un riferimento a una "Python Enhancement Proposal". Genera voci di indice appropriate. Viene generato il testo "PEP *number*" nell'output HTML, questo testo è un link ipertestuale a una copia online del PEP specificato. Ci si può collegare a una sezione specifica dicendo `:pep: `number#anchor``.

Per esempio: [PEP 8](#)¹⁵⁰

:rfc:

Un riferimento a una "Internet Request for Comments". Genera voci di indice appropriate. Viene generato il testo "RFC *number*"; nell'output HTML, questo testo è un link ipertestuale a una copia online della RFC specificata. Ci si può collegare a una sezione specifica dicendo `:rfc: `number#anchor``.

Per esempio: [RFC 2324](#)¹⁵¹

Da notare che non esistono ruoli speciali per includere i link ipertestuali, per questo si può utilizzare il markup reST standard.

Sostituzioni

Il sistema di documentazione fornisce alcune sostituzioni definite per default. Sono impostate nel file di configurazione della build.

|release|

Sostituito dalla release del progetto a cui fa riferimento la documentazione. Si tratta della stringa della versione completa, inclusi i tag alpha/beta/release candidate, ad es. 2.5.2b3. Impostato da `release`.

|version|

Sostituita dalla versione del progetto a cui fa riferimento la documentazione. Questo deve consistere solo delle parti della versione 'major' e 'minor', ad es. 2.5, anche per la versione 2.5.1. Impostato da `version`.

|today|

Sostituito dalla data odierna (la data in cui viene letto il documento) o dalla data impostata nel file di configurazione della build. Normalmente ha il formato April 14, 2007. Impostato da `today_fmt` e da `today`.

|translation progress|

Sostituito dall'avanzamento della traduzione del documento. Questa sostituzione è destinata all'uso da parte dei traduttori di documenti come indicatore dell'avanzamento della traduzione.

¹⁵⁰ <https://peps.python.org/pep-0008/>

¹⁵¹ <https://datatracker.ietf.org/doc/html/rfc2324.html>

Direttive

Come discusso in precedenza, una direttiva è un blocco generico di markup esplicito. Mentre Docutils fornisce una serie di direttive, Sphinx ne fornisce molte di più e utilizza le direttive come uno dei principali meccanismi di estensione.

Consultare *Domini* per i ruoli aggiunti dai domini.

Vedi anche:

Fare riferimento al *reStructuredText Primer* per una panoramica sulle direttive di Docutils.

Indici e Sommari

Poiché reST non dispone di funzionalità per interconnettere più documenti o dividere documenti in più file di output, Sphinx utilizza una direttiva propria per aggiungere relazioni tra i singoli file di cui è composta la documentazione, nonché gli indici [tables of content]. La direttiva `toctree` costituisce l'elemento centrale.

Nota: La semplice "inclusione" di un file in un altro può essere eseguita con la direttiva `include`¹⁵².

Nota: Per creare un sommario [table of content] per il documento corrente (file `.rst`), si usa la direttiva `contents`¹⁵³ standard di reST.

.. toctree::

Questa direttiva inserisce un "albero TOC" (Tabella Dei Contenuti) nella posizione corrente, utilizzando i singoli TOC (compresi gli "alberi sub-TOC") dei documenti inseriti nel corpo della direttiva. I relativi nomi dei documenti (che non iniziano con una barra) sono relativi al documento in cui si trova la direttiva, i nomi assoluti sono relativi alla directory sorgente. Si può fornire un'opzione numerica `maxdepth` per indicare la profondità dell'albero; per default, tutti i livelli sono inclusi.¹⁶⁹

La rappresentazione dell'"albero del sommario" ["TOC tree"] cambia per ciascun formato di output. I builder che generano più file (es. HTML) lo trattano come una raccolta di link ipertestuali. D'altra parte, i builder che producono un singolo file (es. LaTeX, man page, ecc.) lo sostituiscono con il contenuto dei documenti nel [TOC tree].

Si consideri questo esempio (tratto dall'indice di riferimento della libreria dei documenti Python):

```
.. toctree::
   :maxdepth: 2
```

(continues on next page)

¹⁵² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

¹⁵³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

(continua dalla pagina precedente)

```
intro
strings
datatypes
numeric
(many more documents listed here)
```

Ciò realizza due cose:

- Vengono inseriti i sommari [TOC] di tutti questi documenti, con una profondità massima di due, ovvero dei titoli nidificati. Vengono prese in considerazione anche le direttive `toctree` presenti in questi documenti.
- Sphinx conosce l'ordine relativo dei documenti `intro`, `strings` e così via, e sa che sono figli del documento mostrato, l'indice della libreria. Da queste informazioni genera i link "capitolo successivo", "capitolo precedente" e "capitolo principale".

Entries

I titoli dei documenti in `toctree` verranno letti automaticamente dal titolo del documento a cui si fa riferimento. Se non è quello che si vuole, si può specificare un titolo e un target espliciti utilizzando una sintassi simile ai link ipertestuali reST (e la *sintassi dei riferimenti incrociati* di Sphinx). Questo assomiglia a:

```
.. toctree::

    intro
    All about strings <strings>
    datatypes
```

La seconda riga sopra si collegherà al documento `strings`, ma utilizzerà il titolo "Tutto sulle stringhe" invece del titolo del documento `strings`.

Si possono aggiungere anche collegamenti esterni, fornendo un URL HTTP invece del nome di un documento.

Numerazione delle sezioni

Per avere i numeri di sezione anche nell'output HTML, si assegna al `toctree` **toplevel** un'opzione `numbered`. Per esempio:

```
.. toctree::
   :numbered:

    foo
    bar
```

La numerazione inizia quindi dall'intestazione di `foo`. I sotto-alberi [sub-toctree] vengono numerati automaticamente (non assegnare loro il flag `numbered`).

È anche possibile numerare fino ad una profondità specifica, fornendola come argomento numerico a `numbered`.

Opzioni aggiuntive

Si può utilizzare l'opzione `caption` per fornire una didascalia di toctree e si può utilizzare l'opzione `name` per dare un nome al target implicito a cui è possibile fare riferimento utilizzando `ref`:

```
.. toctree::  
   :caption: Table of Contents  
   :name: mastertoc  
  
   foo
```

Per visualizzare solo i titoli dei documenti nell'albero e non altre intestazioni dello stesso livello, si può utilizzare l'opzione `titlesonly`:

```
.. toctree::  
   :titlesonly:  
  
   foo  
   bar
```

Si può usare il "globbing" nelle direttive toctree, fornendo l'opzione flag `glob`. Tutte le voci vengono poi confrontate con l'elenco dei documenti disponibili e le corrispondenze vengono inserite nell'elenco in ordine alfabetico. Esempio:

```
.. toctree::  
   :glob:  
  
   intro*  
   recipe/*  
   *
```

Ciò include prima tutti i documenti i cui nomi iniziano con `intro`, poi tutti quelli nella cartella `recipe`, quindi tutti i documenti rimanenti (tranne quello contenente la direttiva, ovviamente).¹⁷⁰

Il nome della voce speciale `self` indica il documento contenente la direttiva toctree. È utile per generare una "sitemap" dal toctree.

L'opzione `reversed` serve per invertire l'ordine delle voci nell'elenco. Può essere utile quando si usa l'opzione `glob` per invertire l'ordine dei file. Esempio:

```
.. toctree::  
   :glob:  
   :reversed:  
  
   recipe/*
```

C'è anche un'opzione "hidden" per la direttiva, in questo modo:

```
.. toctree::
   :hidden:

   doc_1
   doc_2
```

Verrà notificata comunque a Sphinx la gerarchia del documento, ma verranno inseriti i link nel documento nella posizione della direttiva: questo ha senso se si vogliono inserire tali collegamenti a mano, in uno stile diverso o nella barra laterale HTML.

Per avere un solo toctree di livello superiore e nascondere tutti gli altri di livello inferiore, si aggiunge l'opzione "includehidden" alla voce del toctree di livello superiore:

```
.. toctree::
   :includehidden:

   doc_1
   doc_2
```

Tutte le altre voci del toctree possono poi essere eliminate con l'opzione "hidden".

Alla fine, tutti i documenti nella *source directory* (o le sottodirectory) devono trovarsi in qualche direttiva toctree; Sphinx emetterà un warning se trova un file non incluso, perché ciò significa che tale file non sarà raggiungibile tramite la navigazione standard.

Si usa *exclude_patterns* per escludere esplicitamente documenti o directory dalla build completa. Utilizzare *i metadati "orfani"* per consentire la creazione di un documento, ma avvisare Sphinx che non è raggiungibile tramite un toctree.

Il "root document" (selezionato da *root_doc*) è la "radice" [root] della gerarchia del TOC tree. È utilizzabile come pagina principale della documentazione o come un "indice generale" se non si fornisce l'opzione maxdepth.

Cambiato nella versione 0.3: Aggiunta l'opzione "globbing".

Cambiato nella versione 0.6: Aggiunte le opzioni "numbered" e "hidden", nonché i collegamenti esterni e il supporto per riferimenti "self".

Cambiato nella versione 1.0: Aggiunta l'opzione "titlesonly".

Cambiato nella versione 1.1: Aggiunto l'argomento numerico a "numbered".

Cambiato nella versione 1.2: Aggiunta l'opzione "includehidden".

Cambiato nella versione 1.3: Aggiunte le opzioni "caption" e "name".

¹⁶⁹ Il writer LaTeX fa riferimento solo all'opzione maxdepth della prima direttiva toctree nel documento.

¹⁷⁰ Una nota sulla sintassi del globbing: si possono usare i costrutti standard della shell *, ?, [...] e [!...] con la caratteristica che tutti questi non corrispondono alle barre [slash]. Si può usare un doppio asterisco ** per corrispondere a qualsiasi sequenza di caratteri *inclusi* gli slash.

Nomi speciali

Sphinx si riserva alcuni nomi di documenti per proprio uso; non si devono creare documenti con tali nomi -- causerebbe dei problemi.

I nomi di documenti speciali (e le relative pagine generate) sono:

- `genindex`, `modindex`, `search`

Vengono utilizzati rispettivamente per l'indice generale, l'indice del modulo Python e la pagina di ricerca.

L'indice generale è popolato con le voci provenienti dai moduli, da tutte le *descrizioni degli oggetti* indicizzate e dalle direttive *index*.

L'indice del modulo Python contiene una voce per ciascuna direttiva *py:module*.

La pagina di ricerca contiene un modulo che utilizza l'indice di ricerca JSON generato e JavaScript per eseguire ricerche full-text nei documenti generati per le parole ricercabili; dovrebbe funzionare su tutti i principali browser che supportano JavaScript.

- ogni nome che inizia con `_`

Sebbene Sphinx utilizzi attualmente pochi nomi di questo tipo, non si devono creare documenti o directory contenenti documenti con tali nomi. (Usare `_` come prefisso per una directory di template custom, va bene).

Avvertimento: Attenzione ai caratteri insoliti nei nomi dei file. Alcuni formati potrebbero interpretare questi caratteri in modo inaspettato:

- Non utilizzare i due punti `:` per i formati basati su HTML. I link ad altre parti potrebbero non funzionare.
- Non utilizzare il segno più `+` per il formato ePub. Alcune risorse potrebbero non essere trovate.

Markup a livello di paragrafo

Queste direttive creano paragrafi brevi e possono essere utilizzate sia all'interno di unità informative che di testo normale.

.. note::

Un'informazione particolarmente importante su un'API di cui un utente dovrebbe essere a conoscenza quando utilizza qualunque parte dell'API a cui si riferisce la nota. Il contenuto della direttiva dovrebbe essere scritto in frasi complete e includere tutta la punteggiatura appropriata.

Esempio:


```
.. note::
```

```
This function is not suitable for sending spam e-mails.
```

.. warning::

Un'informazione importante su un'API di cui un utente dovrebbe essere molto consapevole quando utilizza la parte dell'API a cui si riferisce il warning. Il contenuto della direttiva dovrebbe essere scritto in frasi complete e includere tutta la punteggiatura appropriata. Questo è diverso da *note* in quanto è consigliato rispetto a *note* per informazioni riguardanti la sicurezza.

.. versionadded:: version

Questa direttiva documenta la versione del progetto che ha aggiunto la funzionalità descritta alla libreria o all'API del C. Quando si applica a un intero modulo, dovrebbe essere posizionato nella parte superiore della sezione del modulo prima di qualsiasi testo.

Deve essere fornito il primo argomento ed è la versione in questione; si può aggiungere un secondo argomento consistente in una *breve* spiegazione della modifica.

Esempio:

```
.. versionadded:: 2.5
   The *spam* parameter.
```

Si noti che non deve esserci alcuna riga vuota tra l'intestazione della direttiva e la spiegazione; questo per rendere questi blocchi visivamente continui nel markup.

.. versionchanged:: version

Simile a *versionadded*, ma descrive quando e cosa è cambiato in qualche modo nella funzionalità denominata (nuovi parametri, effetti collaterali modificati, ecc.).

.. deprecated:: version

Simile a *versionchanged*, ma descrive quando la funzionalità è stata deprecata. Si può anche fornire una spiegazione, ad esempio per informare il lettore su cosa dovrebbe essere utilizzato al suo posto. Esempio:

```
.. deprecated:: 3.1
   Use :func:`spam` instead.
```

.. versionremoved:: version

Simile a *versionadded*, ma descrive quando la funzionalità è stata rimossa. Potrebbe essere fornita una spiegazione per informare il lettore su cosa utilizzare invece o sul motivo per cui la funzionalità è stata rimossa. Esempio:

```
.. versionremoved:: 4.0
   The :func:`spam` function is more flexible, and should be used_
   ↪ instead.
```

Added in version 7.3.

.. seealso::

Molte sezioni includono un elenco di riferimenti alla documentazione del modulo o a documenti esterni. Questi elenchi vengono creati utilizzando la direttiva *seealso*.

La direttiva *seealso* è tipicamente posizionata in una sezione appena prima di qualsiasi sottosezione. Per l'output HTML, viene visualizzato in un riquadro rispetto al flusso principale del testo.

Il contenuto della direttiva *seealso* dovrebbe essere un elenco di definizioni reST. Esempio:

```
.. seealso::

Module :py:mod:`zipfile`
    Documentation of the :py:mod:`zipfile` standard module.

`GNU tar manual, Basic Tar Format <https://link>`_
    Documentation for tar archive files, including GNU tar_
→extensions.
```

È consentita anche una "forma breve" simile a questa:

```
.. seealso:: modules :py:mod:`zipfile`, :py:mod:`tarfile`
```

Added in version 0.5: La forma breve.

.. rubric:: title

Questa direttiva crea un titolo di paragrafo che non viene utilizzato per creare una voce del sommario.

Nota: Se il *titolo* della rubrica è "Note a piè di pagina" (o l'equivalente nella lingua selezionata), questa rubrica viene ignorata dal writer LaTeX, poiché si presuppone che contenga solo le definizioni delle note a piè di pagina e quindi creerebbe un'intestazione vuota.

.. centered::

Questa direttiva crea una riga di testo centrata in grassetto. Si usa in questo modo:

```
.. centered:: LICENSE AGREEMENT
```

Deprecato dalla versione 1.1: Questa direttiva di sola presentazione è un'eredità delle versioni precedenti. Utilizzare invece una direttiva *rst-class* aggiungendo uno stile appropriato.

.. hlist::

Questa direttiva deve contenere un elenco puntato. Lo trasformerà in un elenco più compatto distribuendo più di un elemento orizzontalmente o riducendo la spaziatura tra gli elementi, a seconda del builder.

Per i builder che supportano la distribuzione orizzontale, esiste un'opzione `columns` che specifica il numero di colonne; il valore di default è 2. Esempio:

```
.. hlist::
   :columns: 3

   * A list of
   * short items
   * that should be
   * displayed
   * horizontally
```

Added in version 0.6.

Mostrare esempi di codice

Ci sono diversi modi per mostrare i blocchi di codice letterale con evidenziazione della sintassi in Sphinx:

- usando *blocchi doctest reST*;
- usando *blocchi letterali reST*, facoltativamente in combinazione con la direttiva *highlight*;
- usando la direttiva *code-block*;
- e usando la direttiva *literalinclude*.

I blocchi doctest sono utilizzabili solo per mostrare sessioni interattive di Python, mentre i restanti tre possono essere utilizzati per altri linguaggi. Di questi tre, i blocchi letterali sono utili quando un intero documento, o almeno ampie sezioni di esso, utilizza blocchi di codice con la stessa sintassi e che dovrebbero avere lo stesso stile. D'altra parte, la direttiva *code-block* ha più senso quando si desidera un controllo più preciso sullo stile di ciascun blocco o quando si ha un documento contenente blocchi di codice che utilizzano più sintassi diverse. Infine, la direttiva *literalinclude* è utile per includere interi file di codice nella documentazione.

In tutti i casi, l'evidenziazione della sintassi è fornita da [Pygments](https://pygments.org)¹⁵⁴. Quando si usano blocchi letterali, viene configurato utilizzando qualsiasi direttiva *highlight* nel file sorgente. Quando viene incontrata una direttiva *highlight*, questa viene utilizzata fino a quando non viene incontrata la successiva direttiva *highlight*. Se nel file non è presente alcuna direttiva *highlight* viene utilizzato il linguaggio di evidenziazione globale. Il valore di default è `python` ma è configurabile col valore di configurazione *highlight_language*. Sono supportati i seguenti valori:

- `none` (nessuna evidenziazione)
- `default` (simile a `python3` ma con una ricaduta su `none` senza che ci sia un warning sulla non visualizzazione; il default quando *highlight_language* non è impostato)
- `guess` (lascia a Pygments scegliere il lexer in base ai contenuti, funziona solo con alcuni linguaggi ben riconoscibili)

¹⁵⁴ <https://pygments.org>

- python
- rest
- c
- ... e qualsiasi altro *alias lexer* supportato da Pygments¹⁵⁵

Se l'evidenziazione col linguaggio selezionato fallisce (cioè Pygments emette un token "Error"), il blocco non viene evidenziato affatto.

Importante: L'elenco degli *alias lexer* supportati è legato alla versione Pygment. Volendo garantire un'evidenziazione coerente, si deve correggere la versione di Pygments.

.. highlight:: language

Esempio:

```
.. highlight:: c
```

Questo linguaggio viene utilizzato finché non viene incontrata la successiva direttiva `highlight`. Come discusso in precedenza, *language* può essere qualsiasi *alias lexer* supportato da Pygments.

opzioni

:linenothreshold: threshold (number (optional))

Abilita la generazione di numeri di riga per i blocchi di codice.

Questa opzione accetta un numero facoltativo come parametro di soglia. Se c'è una soglia [threshold], la direttiva produrrà numeri di riga solo per i blocchi di codice più lunghi di N righe. Altrimenti, verranno prodotti i numeri di riga per tutti i blocchi di codice.

Esempio:

```
.. highlight:: python
:linenothreshold: 5
```

:force: (no value)

Se fornito, gli errori minori nell'evidenziazione vengono ignorati.

Added in version 2.1.

.. code-block:: [language]

.. sourcecode:: [language]

Esempio:

¹⁵⁵ <https://pygments.org/docs/lexers>

```
.. code-block:: ruby
```

```
Some Ruby code.
```

Funziona anche il nome dell'alias della direttiva `sourcecode`. Questa direttiva accetta il nome di un linguaggio come argomento. Può essere qualsiasi alias di `lexer supportato da Pygments`¹⁵⁶. Se non viene fornito, verrà utilizzata l'impostazione della direttiva `highlight`. Se non impostato, verrà utilizzato `highlight_language`. Per visualizzare un esempio di codice *inline* all'interno di un altro testo, anziché come un blocco separato, si può invece utilizzare il ruolo `code`.

Cambiato nella versione 2.0: L'argomento `language` diventa facoltativo.

opzioni

:linenos: (no value)

Abilita la generazione di numeri di riga per il blocco di codice:

```
.. code-block:: ruby
   :linenos:
```

```
Some more Ruby code.
```

:lineno-start: number (number)

Imposta il numero della prima riga del blocco di codice. Se presente, viene attivata anche l'opzione `linenos`:

```
.. code-block:: ruby
   :lineno-start: 10
```

```
Some more Ruby code, with line numbering starting at 10.
```

Added in version 1.3.

:emphasize-lines: line numbers (comma separated numbers)

Enfatizza righe particolari del blocco di codice:

```
.. code-block:: python
   :emphasize-lines: 3,5

   def some_function():
       interesting = False
       print('This line is highlighted.')
       print('This one is not...')
       print('...but this one is.')
```

Added in version 1.1.

Cambiato nella versione 1.6.6: LaTeX supporta l'opzione `emphasize-lines`.

:caption: caption of code block (text)

Imposta una didascalia per il blocco di codice.

Added in version 1.3.

:name: a label for hyperlink (text)

Definisce il nome del target implicito a cui è possibile fare riferimento utilizzando *ref*. Per esempio:

```
.. code-block:: python
   :caption: this.py
   :name: this-py

   print('Explicit is better than implicit.')
```

Per creare riferimenti incrociati ad un blocco di codice utilizzando il ruolo *ref* o *numref*, è necessario che sia **name** che **caption**. L'argomento di **name** può poi essere fornito a *numref* per generare il riferimento incrociato. Esempio:

```
See :numref:`this-py` for an example.
```

Quando si usa *ref*, è possibile generare un riferimento incrociato la sola definizione di **name**, a condizione che venga fornito un titolo esplicito. Esempio:

```
See :ref:`this code snippet <this-py>` for an example.
```

Added in version 1.3.

:class: class names (a list of class names separated by spaces)

Il nome della classe del grafico.

Added in version 1.4.

:dedent: number (number or no value)

Elimina i caratteri di indentazione dal blocco di codice. Quando viene fornito il numero, gli N caratteri iniziali vengono rimossi. Quando non viene fornito alcun argomento, gli spazi iniziali vengono rimossi tramite `textwrap.dedent()`¹⁵⁷. Per esempio:

```
.. code-block:: ruby
   :linenos:
   :dedent: 4

   some ruby code
```

Added in version 1.3.

Cambiato nella versione 3.5: Supporto della de-indentazione automatica.

:force: (no value)

Se fornito, gli errori minori nell'evidenziazione vengono ignorati.

Added in version 2.1.

.. literalinclude:: filename

È possibile includere visualizzazioni più lunghe di testo letterale memorizzando il testo di esempio in un file esterno contenente solo testo semplice. Il file può essere incluso utilizzando la direttiva `literalinclude`.¹⁷¹ Ad esempio, per includere il file sorgente Python `example.py`, si usa:

```
.. literalinclude:: example.py
```

Il nome del file è solitamente relativo al path del file corrente. Tuttavia, se è assoluto (iniziando con `/`), è relativo alla directory sorgente principale.

Opzioni aggiuntive

Come *code-block*, la direttiva supporta l'opzione flag `linenos` per attivare i numeri di riga, l'opzione `lineno-start` per selezionare il numero della prima riga, l'opzione `emphasize-lines` per enfatizzare righe particolari, l'opzione `name` per fornire un nome di target implicito, l'opzione `dedent` per rimuovere i caratteri di rientro per il blocco di codice e un'opzione `language` per selezionare un linguaggio diverso da quello standard del file corrente. Inoltre, supporta l'opzione `caption`; tuttavia, è possibile fornirlo senza argomenti per utilizzare il nome del file come didascalia. Esempi con opzioni:

```
.. literalinclude:: example.rb
   :language: ruby
   :emphasize-lines: 12,15-18
   :linenos:
```

Le schede nell'input vengono espanse se si fornisce un'opzione `tab-width` con la larghezza desiderata della scheda.

Si presuppone che i file include siano codificati in *source_encoding*. Se il file ha una codifica diversa, la si può specificare con l'opzione `encoding`:

```
.. literalinclude:: example.py
   :encoding: latin-1
```

La direttiva supporta anche l'inclusione solo di parti del file. Se si tratta di un modulo Python, si può selezionare una classe, una funzione o un metodo da includere utilizzando l'opzione `pyobject`:

```
.. literalinclude:: example.py
   :pyobject: Timer.start
```

Ciò includerebbe solo le righe di codice appartenenti al metodo `start()` nella classe `Timer` nel file.

In alternativa, si può specificare esattamente quali righe includere fornendo un'opzione `lines`:

¹⁵⁶ <https://pygments.org/docs/lexers/>

¹⁵⁷ <https://docs.python.org/3/library/textwrap.html#textwrap.dedent>

```
.. literalinclude:: example.py
   :lines: 1,3,5-10,20-
```

Questo include le righe 1, 3, da 5 a 10 e dalla riga 20 fino alla fine.

Un altro modo per controllare quale parte del file includere è quello di utilizzare le opzioni `start-after` e `end-before` (o solo una di esse). Se `start-after` viene fornito come opzione di stringa, vengono incluse solo le righe che seguono la prima riga contenente quella stringa. Se `end-before` viene fornito come opzione di stringa, vengono incluse solo le righe che precedono le prime righe contenenti quella stringa. Le opzioni `start-at` e `end-at` si comportano in modo simile, ma vengono incluse le righe contenenti la stringa corrispondente.

`start-after/start-at` e `end-before/end-at` possono avere la stessa stringa. `start-after/start-at` filtrano le righe prima della riga che contiene la stringa dell'opzione (`start-at` manterrà la riga). Quindi `end-before/end-at` filtrano le righe dopo la riga che contiene la stringa dell'opzione (`end-at` manterrà la riga e `end-before` salta la prima riga).

Nota: Se si vuole selezionare solo `[second-section]` del file ini come il seguente, si possono usare `:start-at: [second-section]` e `:end-before: [third-section]`:

```
[first-section]

var_in_first=true

[second-section]

var_in_second=true

[third-section]

var_in_third=true
```

Casi utili per queste opzioni sono per gestire i commenti dei tag. Le opzioni `:start-after: [initialize]` e `:end-before: [initialized]` mantengono le righe tra i commenti:

```
if __name__ == "__main__":
    # [initialize]
    app.start(":8000")
    # [initialized]
```

Quando le righe sono state selezionate in uno dei modi descritti sopra, i numeri di riga in `emphasize-lines` si riferiscono a tali righe selezionate, contate consecutivamente a partire da 1.

Quando si specificano porzioni particolari di un file da visualizzare, può essere utile mostrare i numeri di riga originali. Questo può essere fatto utilizzando l'opzione `lineno-match`, che è però consentita solo quando la selezione è composta da righe contigue.

Si può anteporre e/o aggiungere una riga al codice incluso, utilizzando rispettivamente le opzioni `prepend` e `append`. Questo è utile ad es. per evidenziare il codice PHP che non include i marcatori `<?php/?>`.

Per mostrare il diff del codice, si può specificare il vecchio file fornendo un'opzione `diff`:

```
.. literalinclude:: example.py
   :diff: example.py.orig
```

Questo mostra la differenza tra `example.py` e `example.py.orig` con il formato diff unificato.

Con l'opzione `force` si possono ignorare gli errori minori nell'evidenziazione.

Cambiato nella versione 0.4.3: Aggiunta l'opzione `encoding`.

Cambiato nella versione 0.6: Aggiunte le opzioni `pyobject`, `lines`, `start-after` e `end-before`, nonché supporto per 'nomi di file' assoluti.

Cambiato nella versione 1.0: Aggiunte le opzioni `prepend`, `append` e `tab-width`.

Cambiato nella versione 1.3: Aggiunte le opzioni `diff`, `lineno-match`, `caption`, `name` e `dedent`.

Cambiato nella versione 1.4: Aggiunta l'opzione `class`.

Cambiato nella versione 1.5: Aggiunte le opzioni `start-at` e `end-at`.

Cambiato nella versione 1.6: Sia utilizzando `start-after` che `lines`, la prima riga per `start-after` è considerata con il numero di riga 1 per ogni `lines`.

Cambiato nella versione 2.1: Aggiunta l'opzione `force`.

Cambiato nella versione 3.5: Supporto della de-indentazione automatica.

Glossario

.. glossary::

Questa direttiva deve contenere un markup simile a un elenco di definizioni reST con termini e definizioni. Le definizioni saranno poi referenziabili con il ruolo `term`. Esempio:

```
.. glossary::

    environment
```

(continues on next page)

¹⁷¹ Esiste una direttiva standard `.. include`, ma genera errori se il file non viene trovato. Questo emette solo un warning.

(continua dalla pagina precedente)

```

A structure where information about all documents under the_
→root is
    saved, and used for cross-referencing. The environment is_
→pickled
    after the parsing stage, so that successive runs only need_
→to read
    and parse new and changed documents.

source directory
The directory which, including its subdirectories, contains_
→all
    source files for one Sphinx project.

```

A differenza dei normali elenchi di definizioni, sono consentiti *più* termini per ogni voce e il markup in linea è consentito nei termini. È possibile avere link a tutti i termini. Per esempio:

```

.. glossary::

    term 1
    term 2
    Definition of both terms.

```

(Quando il glossario è ordinato, il primo termine determina l'ordinamento).

Se si vuole specificare la "chiave di raggruppamento" [grouping key] per le voci dell'indice generale, si può inserire una "chiave" come in "term : chiave". Per esempio:

```

.. glossary::

    term 1 : A
    term 2 : B
    Definition of both terms.

```

Da notare che la "chiave" viene utilizzata così com'è per i raggruppamenti. La "chiave" non è normalizzata; le chiavi "A" e "a" diventano gruppi diversi. Vengono utilizzati tutti i caratteri in "chiave", non solo il primo; viene utilizzato per la chiave di raggruppamento "Combining Character Sequence" e per "Surrogate Pairs".

Nelle internazionalizzazioni [i18n], si possono specificare "termine localizzato : chiave" anche se il testo originale contiene solo la parte "termine". In questo caso, il "termine localizzato" tradotto verrà classificato nel gruppo "chiave".

Added in version 0.6: Ora si può dare alla direttiva del glossario un flag `:sorted:` che ordinerà automaticamente le voci in ordine alfabetico.

Cambiato nella versione 1.1: Ora supporta più termini e markup in linea nei termini.

Cambiato nella versione 1.4: La chiave dell'indice per i termini del glossario deve essere considerata *sperimentale*.

Cambiato nella versione 4.4: Nella documentazione internazionalizzata, il flag `:sorted:` ordina in base ai termini tradotti.

Markup di meta-informazioni

.. sectionauthor:: name <email>

Identifica l'autore della sezione corrente. L'argomento dovrebbe includere il nome dell'autore in modo che possa essere utilizzato per la presentazione e l'indirizzo e-mail. La parte del nome del dominio dell'indirizzo deve essere in minuscolo. Esempio:

```
.. sectionauthor:: Guido van Rossum <guido@python.org>
```

Per default, questo markup non si riflette in alcun modo nell'output (aiuta a tenere traccia dei contributi), ma si può impostare il valore di configurazione `show_authors` a `True` per fargli produrre un paragrafo nell'output.

.. codeauthor:: name <email>

La direttiva `codeauthor`, che può apparire più volte, nomina gli autori del codice descritto, proprio come `sectionauthor` descrive il/gli autore/i di una porzione di documentazione. Anch'esso produce un output solo se il valore di configurazione `show_authors` è `True`.

Markup che generano indici

Sphinx crea automaticamente voci di indice da tutte le descrizioni degli oggetti (come funzioni, classi o attributi) come discusso in *Domini*.

Tuttavia, è disponibile anche un markup esplicito, per rendere l'indice più completo e abilitare le voci dell'indice nei documenti in cui le informazioni non sono principalmente contenute in unità informative, come il 'reference' del linguaggio.

.. index:: <entries>

Questa direttiva contiene una o più voci di indice. Ogni voce è composta da un tipo e un valore, separati da due punti.

Per esempio:

```
.. index::
    single: execution; context
    pair: module; __main__
    pair: module; sys
    triple: module; search; path
    seealso: scope
```

The execution context

...

Questa direttiva contiene cinque voci, che verranno convertite in voci nell'indice generato che si collegano alla posizione esatta della dichiarazione dell'indice (o, in caso di media offline, al numero di pagina corrispondente).

Poiché le direttive dell'indice generano destinazioni di riferimenti incrociati nella loro posizione nell'origine, ha senso metterle *prima* dell'oggetto a cui si riferiscono, ad es. un titolo, come nell'esempio sopra.

I tipi di voci possibili sono:

single

Crea una singola voce di indice. È possibile creare una voce secondaria separando il testo della voce secondaria con un punto e virgola (questa notazione viene utilizzata anche di seguito per descrivere quali voci creare). Esempi:

```
.. index:: single: execution
           single: execution; context
```

- `single: execution` crea una voce di indice etichettata con `execution`.
- `single: execution; context` crea una sottovoce di `execution` etichettata con `context`.

pair

Una scorciatoia per creare due voci di indice. La coppia di valori deve essere separata da un punto e virgola. Esempio:

```
.. index:: pair: loop; statement
```

Questo crea due voci di indice; `loop; statement` e `statement; loop`.

triple

Una scorciatoia per creare tre voci di indice. Tutti e tre i valori devono essere separati da un punto e virgola. Esempio:

```
.. index:: triple: module; search; path
```

Crea tre voci di indice; `module; search path`, `search; path`, `module e path; module search`.

see

Una scorciatoia per creare una voce di indice che fa riferimento a un'altra voce. Esempio:

```
.. index:: see: entry; other
```

Crea una voce di indice che fa riferimento da `entry` a `other` (cioè 'entry': Vedere 'other').

seealso

Come `see`, ma inserisce 'Vedi anche' anziché 'Vedi'.

module, keyword, operator, object, exception, statement, builtin

Queste scorciatoie **deprecate** creano tutte due voci di indice. Ad esempio, `module:` `hashlib` crea le voci `module;` `hashlib` e `hashlib; module`.

Deprecato dalla versione 1.0: Questi tipi di voci specifiche di Python sono deprecati.

Cambiato nella versione 7.1: Rimozione dell'impostazione della versione a Sphinx 9.0. L'utilizzo di questi tipi di voci ora genererà warning con la categoria `index`.

È possibile contrassegnare le voci dell'indice "main" prefissandole con un punto esclamativo. I riferimenti alle voci "main" sono enfatizzati nell'indice generato. Ad esempio, se due pagine contengono:

```
.. index:: Python
```

e una pagina contiene:

```
.. index:: ! Python
```

allora tra i tre backlink viene enfatizzato quello a quest'ultima pagina.

Per le direttive dell'indice contenenti solo voci "singole", esiste una notazione abbreviata:

```
.. index:: BNF, grammar, syntax, notation
```

Vengono create quattro voci di indice.

Cambiato nella versione 1.1: Aggiunti i tipi `see` e `seealso`, oltre a contrassegnare le voci 'main'.

opzioni**:name: a label for hyperlink (text)**

Definisce il nome del target implicito a cui è possibile fare riferimento utilizzando *ref*. Per esempio:

```
.. index:: Python
   :name: py-index
```

Added in version 3.0.

:index:

Mentre la direttiva *index* è un markup a livello di blocco e si collega all'inizio del paragrafo successivo, esiste anche un ruolo corrispondente che imposta la destinazione del collegamento direttamente dove viene utilizzato.

Il contenuto del ruolo può essere una semplice frase, che viene poi mantenuta nel testo e utilizzata come voce di indice. Può anche essere una combinazione di testo e voce di indice, in stile con target espliciti di riferimenti incrociati. In tal caso, la parte "target" può essere una voce completa come descritto per la direttiva sopra. Per esempio:

This is a normal reST `:index: `paragraph`` that contains several `:index: `index entries <pair: index; entry>``.

Added in version 1.1.

Inclusione di contenuto basato sui tag

`.. only:: <expression>`

Include il contenuto della direttiva solo se *expression* è vera [true]. *expression* dovrebbe essere composta da tag, come questo:

```
.. only:: html and draft
```

I tag non definiti sono 'false', quelli definiti (tramite l'opzione della riga di comando `-t` o all'interno di `conf.py`, vedere *qui*) sono 'true'. Sono supportate le espressioni booleane, che utilizzano anche le parentesi (come `(latex or html)` and `draft`).

Il *format* e il *name* del builder corrente (`html`, `latex` o `text`) sono sempre impostati come tag¹⁷². Per rendere esplicita la distinzione tra *format* e *name*, vengono aggiunti anche i prefissi `format_` e `builder_`, ad es. il builder `epub` definisce i tag `html`, `epub`, `format_html` e `builder_epub`.

Questi tag standard vengono impostati *dopo* che il file di configurazione è stato letto, quindi non sono disponibili lì.

Tutti i tag devono seguire la sintassi standard degli identificatori Python come stabilito nella documentazione *Identifiers and keywords*¹⁵⁸. Cioè, un'espressione tag può consistere solo di tag conformi alla sintassi delle variabili Python. In ASCII, è composto dalle lettere maiuscole e minuscole da A fino alla Z, l'underscore `_` e, ad eccezione del primo carattere, dalle cifre da 0 a 9.

Added in version 0.6.

Cambiato nella versione 1.2: Aggiunto il nome del builder e i prefissi.

Avvertimento: Questa direttiva è progettata per controllare solo il contenuto del documento. Non poteva controllare sezioni, etichette e così via.

¹⁷² Per la maggior parte dei builder il nome e il formato sono gli stessi. Al momento solo i builder derivati da quello `html` fanno distinzione tra il formato e il nome del builder.

Si tenga presente che il tag del builder corrente non è disponibile in `conf.py`, poiché viene creato dopo che il builder è stato inizializzato.

¹⁵⁸ https://docs.python.org/3/reference/lexical_analysis.html#identifiers

Tabelle

Utilizzare le *tabelle reStructuredText*, ovvero

- la sintassi della tabella griglia ([ref¹⁵⁹](#)),
- la sintassi della tabella semplice ([ref¹⁶⁰](#)),
- la sintassi `csv-table`¹⁶¹,
- o la sintassi `list-table`¹⁶².

La direttiva `table`¹⁶³ serve come wrapper opzionale delle sintassi *grid* e *simple*.

Funzionano bene nell'output HTML, ma il rendering delle tabelle in LaTeX è complicato. Controllare la `latex_table_style`.

Cambiato nella versione 1.6: Unione di celle (multi-riga, multi-colonna, entrambe) da tabelle a griglia con contenuti complessi come più paragrafi, virgolette, elenchi, blocchi letterali, verranno visualizzate correttamente nell'output LaTeX.

`.. tabularcolumns:: column spec`

Questa direttiva influenza solo l'output LaTeX per la tabella successiva nel sorgente. L'argomento obbligatorio è una specifica di colonna (nota come "alignment preamble" (preambolo di allineamento) nel gergo LaTeX). Si faccia riferimento alla documentazione LaTeX, come la [wiki page](#)¹⁶⁴, per le nozioni di base su tale specifica di colonna.

Added in version 0.3.

Nota: `tabularcolumns` è in conflitto con l'opzione `:widths:` delle direttive della tabella. Se vengono specificati entrambi, l'opzione `:widths:` verrà ignorata.

Sphinx eseguirà il rendering delle tabelle con più di 30 righe con `longtable`. Oltre agli identificatori di colonna `l`, `r`, `c` e `p{width}`, è possibile utilizzare anche `\X{a}{b}` (nuovo nella versione 1.5) che configura la larghezza della colonna come una frazione a/b della larghezza totale della riga e `\Y{f}` (nuovo nella versione 1.6) dove f è un decimale: ad esempio `\Y{0.2}` significa che la colonna occuperà 0.2 volte la larghezza della riga.

Quando questa direttiva viene utilizzata per una tabella con un massimo di 30 righe, Sphinx la rappresenterà con `tabulary`. È poi possibile utilizzare tipi di colonna specifici `L` (left), `R` (right), `C` (centered) e `J` (justified). Hanno l'effetto di una `p{width}` (ovvero ogni cella è un `\parbox` LaTeX) con l'allineamento interno del testo specificato e una `width` (larghezza) calcolata automaticamente.

Avvertimento:

¹⁵⁹ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#grid-tables>

¹⁶⁰ <https://docutils.sourceforge.io/docs/ref/rst/restructuredtext.html#simple-tables>

¹⁶¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#csv-table>

¹⁶² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#list-table>

¹⁶³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table>

- Le celle che contengono elementi simili a elenchi come descrizioni di oggetti, virgolette o qualsiasi tipo di elenco non sono compatibili con i tipi di colonna LRCJ. Il tipo di colonna deve quindi essere `p{width}` con una `width` esplicita (o `\X{a}{b}` o `\Y{f}`).
- I blocchi letterali non funzionano affatto con `tabulary`. Sphinx tornerà agli ambienti `tabular` o `longtable` e genererà una specifica di colonna adatta.

In assenza della direttiva `tabularcolumns` e per una tabella con un massimo di 30 righe e nessuna cella problematica come descritto nel warning precedente, Sphinx utilizza `tabulary` e il tipo di colonna `J` per ogni colonna.

Cambiato nella versione 1.6: In precedenza veniva utilizzato il tipo di colonna `L` (il testo è allineato a sinistra). Per tornare a questo, si deve includere `\newcolumnntype{T}{L}` nel preambolo LaTeX, poiché in effetti Sphinx usa `T` e lo imposta di default come alias di `J`.

Suggerimento: Un problema frequente con `tabulary` è che le colonne con poco contenuto sembrano "schiacciate". Si può aggiungere al preambolo LaTeX, `\setlength{\tymin}{40pt}` ad esempio, per garantire una larghezza minima della colonna di `40pt`, il `tabulary` con un default di `10pt` è troppo piccolo.

Suggerimento: Per forzare l'utilizzo di `longtable` dell'ambiente LaTeX si passa `longtable` come opzione `:class:` a `table`¹⁶⁵, `csv-table`¹⁶⁶, o a `list-table`¹⁶⁷. Si usa `rst-class` per altre tabelle.

Formule matematiche

Il linguaggio di input per la matematica è il markup LaTeX. Questo è lo standard de facto per la notazione matematica in testo semplice e presenta l'ulteriore vantaggio che non è necessaria alcuna ulteriore traduzione durante la creazione dell'output LaTeX.

Da tener presente che quando si inserisce il markup `math` nelle **docstring Python** lette da `autodoc`, bisogna raddoppiare tutte le backslash, o utilizzare le raw string di Python (`r"raw"`).

.. math::

Direttiva per visualizzare le formule matematiche (formule che prendono per sé l'intera riga).

La direttiva supporta più equazioni, che dovrebbero essere separate da una riga vuota:

¹⁶⁴ <https://en.wikibooks.org/wiki/LaTeX/Tables>

¹⁶⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table>

¹⁶⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#csv-table>

¹⁶⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#list-table>


```
.. math::
```

$$(a + b)^2 = a^2 + 2ab + b^2$$

$$(a - b)^2 = a^2 - 2ab + b^2$$

Inoltre, ogni singola equazione è impostata all'interno di un ambiente `split`, il che significa che si possono avere più righe allineate in un'equazione, allineate a & e separate da `\\`:

```
.. math::
```

$$\begin{aligned} (a + b)^2 &= (a + b)(a + b) \\ &= a^2 + 2ab + b^2 \end{aligned}$$

Per maggiori dettagli, consultare la documentazione del [pacchetto AmSMath LaTeX](#)¹⁶⁸.

Quando la formula è solo una riga di testo, può anche essere fornita da un argomento di una direttiva:

```
.. math:: (a + b)^2 = a^2 + 2ab + b^2
```

Normalmente le equazioni non sono numerate. Se si vuole la numerazione, si usa l'opzione `label`. Quando viene fornito, seleziona un'etichetta interna per l'equazione, tramite la quale è possibile fare riferimenti incrociati, e provoca l'emissione di un numero di equazione. Consultare [eq](#) per un esempio. Lo stile della numerazione dipende dal formato dell'output.

C'è anche un'opzione `nowrap` che impedisce qualsiasi ritorno a capo della formula in un ambiente matematico. Dando questa opzione, si deve impostare correttamente la formula. Per esempio:

```
.. math::
```

```
:nowrap:
```

$$\begin{aligned} \backslash\mathrm{begin}\{\mathrm{eqnarray}\} \\ y \quad &= & ax^2 + bx + c \\ f(x) &= & x^2 + 2xy + y^2 \\ \backslash\mathrm{end}\{\mathrm{eqnarray}\} \end{aligned}$$

Vedi anche:

Supporto matematico per output HTML in Sphinx

Opzioni di rendering per le formule con i builder HTML.

latex_engine

Spiega come configurare il builder LaTeX per supportare i letterali Unicode nei markup delle formule.

¹⁶⁸ <https://www.ams.org/publications/authors/tex/amslatex>

Visualizzazioni della produzione grammaticale

È disponibile un markup speciale per visualizzare le produzioni di una grammatica formale. Il markup è semplice e non tenta di modellare tutti gli aspetti del BNF (o qualsiasi forma derivata), ma fornisce abbastanza per consentire la visualizzazione di grammatiche libere dal contesto in modo tale da rendere gli usi di un simbolo come i link ipertestuali alla definizione del simbolo. C'è questa direttiva:

.. productionlist:: [productionGroup]

Questa direttiva viene utilizzata per racchiudere un gruppo di produzioni. Ogni produzione è riportata su un'unica riga ed è composta da un nome, separato da due punti dalla definizione seguente. Se la definizione si estende su più righe, ciascuna riga di continuazione deve iniziare con i due punti posizionati nella stessa colonna della prima riga. Non sono consentite righe vuote negli argomenti della direttiva `productionlist`.

La definizione può contenere nomi di token contrassegnati come testo interpretato (ad esempio, `sum ::= `integer` "+" `integer``) -- questo genera riferimenti incrociati alle produzioni di questi token. Al di fuori delle liste di produzione, si può fare riferimento alle produzioni di token utilizzando *token*.

L'argomento *productionGroup* di *productionlist* serve a distinguere i diversi insiemi di liste di produzione che appartengono a grammatiche diverse. Più elenchi di produzione con lo stesso *productionGroup* definiscono quindi delle regole nello stesso "scope".

All'interno delle liste di produzione, i token si riferiscono implicitamente alle produzioni del gruppo corrente. Si può fare riferimento alla produzione di un'altra grammatica anteponendo al token il nome del gruppo e i due punti, ad esempio `otherGroup:sum`. Se il gruppo del token non deve apparire nella produzione, può essere preceduto da una tilde, ad esempio `~otherGroup:sum`. Per fare riferimento a una produzione da una grammatica anonima, il token dovrebbe essere preceduto da due punti, ad esempio `:sum`.

Al di fuori della lista di produzione, se è stato fornito un argomento *productionGroup* si deve anteporre al nome del token nel riferimento incrociato il nome del gruppo e i due punti, ad esempio `myGroup:sum` anziché il solo `sum`. Se il gruppo non deve apparire nel titolo del link, è possibile fornire un titolo esplicito (ad esempio, `myTitle <myGroup:sum>`) oppure è possibile prefissare il target con una tilde (ad esempio, `~myGroup:sum`).

Da tener presente che nella produzione non viene eseguita alcuna ulteriore analisi reST, quindi non è necessario eseguire l'escape dei caratteri `*` e `|`.

Quello che segue è un esempio tratto dal Manuale di riferimento Python:

```
.. productionlist::
try_stmt: try1_stmt | try2_stmt
try1_stmt: "try" ":" `suite`
           : ("except" [ `expression` ["," `target`] ] ":" `suite`)+
           : ["else" ":" `suite`]
           : ["finally" ":" `suite`]
try2_stmt: "try" ":" `suite`
           : "finally" ":" `suite`
```

Liste di campi

Come visto in precedenza, le Field List (Liste di Campi) sono sequenze di campi contrassegnate in questo modo:

```
:fieldname: Field content
```

Sphinx estende il comportamento standard dei documenti per le liste di campi e aggiunge alcune funzionalità extra trattate in questa sezione.

Nota: I valori delle liste di campi verranno analizzati come stringhe. Non è possibile utilizzare collection Python come elenchi o dizionari.

Metadati a livello di file

Una lista di campi nella parte superiore di un file viene normalmente analizzata da docutils come *docinfo* e mostrata nella pagina. Tuttavia, in Sphinx, una lista di campi che precede qualsiasi altro markup viene spostata da *docinfo* all'ambiente Sphinx come metadati del documento e non viene visualizzata nell'output.

Nota: Una lista di campi che appare dopo il titolo del documento *farà* parte del *docinfo* come di consueto e verrà visualizzata nell'output.

Campi di metadati speciali

Sphinx fornisce un comportamento personalizzato per i campi bibliografici rispetto a docutils.

Al momento vengono riconosciuti questi campi di metadati:

tocdepth

La profondità massima per un sommario di questo file.

```
:tocdepth: 2
```

Nota: Questi metadati influiscono sulla profondità del toctree locale. Ma ciò non influisce sulla profondità del toctree *global*. Quindi questo non cambia la barra laterale dei temi che utilizzano il toctree globale.

Added in version 0.4.

nocomments

Se impostato, l'applicazione web non visualizzerà un modulo di commento per una pagina generata da questo file sorgente.

`:nocomments:`

orphan

Se impostato, i warning relativi al fatto che questo file non sarà incluso in nessun toctree verranno soppressi.

`:orphan:`

Added in version 1.0.

nosearch

Se impostato, la ricerca 'full text' per questo file è disabilitata.

`:nosearch:`

Nota: la ricerca dell'oggetto è ancora disponibile anche se è impostata l'opzione `nosearch`.

Added in version 3.0.

SPOSTATO: Domini

SPOSTATO: Basic Markup

Vedere *Domini*.

SPOSTATO: Dominio Python

Vedere *Il Dominio Python*.

SPOSTATO: Dominio C

Vedere *Il Dominio C*.

SPOSTATO: Dominio C++

Vedere *Il Dominio C++*.

SPOSTATO: Dominio Standard

Vedere *Il Dominio Standard*.

SPOSTATO: Dominio JavaScript

Vedere *Il Dominio JavaScript*.

SPOSTATO: Dominio reStructuredText

Vedere *Il Dominio reStructuredText*.

SPOSTATO: Dominio Math

Vedere *Il dominio della Matematica*.

SPOSTATO: Altri domini

Vedere *Domini*.

4.2 Markdown

Markdown¹⁷³ è un linguaggio di markup leggero con una sintassi di formattazione semplice del normale testo. Esiste in molte *varianti* sintatticamente diverse. Per supportare la documentazione basata su Markdown, Sphinx può utilizzare **MyST-Parser**¹⁷⁴. MyST-Parser è un bridge Docutils per **markdown-it-py**¹⁷⁵, un pacchetto Python per il parsing della variante **CommonMark**¹⁷⁶ di Markdown.

Configurazione

Per configurare il progetto Sphinx per supportare il Markdown, si procede come segue:

1. Si installa il parser Markdown *MyST-Parser*:

```
pip install --upgrade myst-parser
```

2. Add *myst_parser* all'*elenco delle estensioni configurate*:

¹⁷³ <https://daringfireball.net/projects/markdown/>

¹⁷⁴ <https://myst-parser.readthedocs.io/en/latest/>

¹⁷⁵ <https://github.com/executablebooks/markdown-it-py>

¹⁷⁶ <https://commonmark.org/>

```
extensions = ['myst_parser']
```

Nota: MyST-Parser richiede Sphinx 2.1 o più recente.

3. Per utilizzare file Markdown con estensioni diverse da `.md`, si modifica la variabile `source_suffix`. L'esempio seguente configura Sphinx per analizzare tutti i file con le estensioni `.md` e `.txt` come Markdown:

```
source_suffix = {
    '.rst': 'restructuredtext',
    '.txt': 'markdown',
    '.md': 'markdown',
}
```

4. Si può ulteriormente configurare *MyST-Parser* per consentire la sintassi personalizzata che lo standard *CommonMark* non supporta. Maggiori informazioni in [MyST-Parser documentation](#)¹⁷⁷.

4.3 Sintassi dei riferimenti incrociati

I riferimenti incrociati sono generati da molti ruoli di testo interpretati semanticamente. Fondamentalmente, si deve solo scrivere `:role:`target``, e verrà creato un link all'elemento chiamato *target* del tipo indicato da *role*. Il testo del link sarà lo stesso di quello del *target*.

Esistono tuttavia alcune funzionalità aggiuntive che rendono i ruoli del riferimento incrociato più versatili:

- Si può fornire un titolo esplicito e un target di riferimento, come nei link ipertestuali reST diretti: `:role:`title <target>`` si riferirà a *target*, ma il testo del link sarà *title*.
- Se si aggiunge al contenuto il prefisso `!`, non verrà creato alcun riferimento/link ipertestuale.
- Se al contenuto si aggiunge il prefisso `~`, il testo del link sarà solo l'ultimo componente del target. Per esempio, `:py:meth:`~Queue.Queue.get`` farà riferimento a `Queue.Queue.get` ma visualizzerà solo `get` come testo del link. Questo non funziona con tutti i ruoli dei riferimenti incrociati, ma è specifico del dominio.

Nell'output HTML, l'attributo `title` del link (che ad esempio viene mostrato come suggerimento [tool-tip] al passaggio del mouse) sarà sempre il nome completo del target.

¹⁷⁷ <https://myst-parser.readthedocs.io/en/latest/using/syntax-optional.html>

Riferimenti incrociati a qualsiasi cosa

:any:

Added in version 1.3.

Questo ruolo di convenienza cerca di fare del suo meglio per trovare un target valido per il suo testo di riferimento.

- Innanzitutto, prova i target dei riferimenti incrociati standard a cui farebbero riferimento *doc*, *ref* o *option*.

Vengono cercati anche gli oggetti custom aggiunti al dominio standard dalle estensioni (vedere `Sphinx.add_object_type()`).

- Poi, cerca gli oggetti (target) in tutti i domini caricati. Spetta ai domini il come debba essere specifica una corrispondenza. Ad esempio, nel dominio Python un riferimento di `:any:`Builder`` corrisponderebbe alla classe `sphinx.builders.Builder`.

Se non viene trovato nessuno o più target, verrà emesso un warning. In caso di target multipli, è possibile modificare "any" in un ruolo specifico.

Questo ruolo è un buon candidato per impostare *default_role*. Facendolo, si possono scrivere riferimenti incrociati senza troppa fatica dei markup. Ad esempio, in questa documentazione della funzione Python:

```
.. function:: install()
```

```
    This function installs a `handler` for every signal known by the
    `signal` module. See the section `about-signals` for more
    ↪information.
```

potrebbero esserci riferimenti a un termine del glossario (solitamente `:term:`handler``), un modulo Python (di solito `:py:mod:`signal`` o `:mod:`signal``) e una sezione (solitamente `:ref:`about-signals``).

Il ruolo *any* funziona anche con l'estensione *intersphinx*: quando non viene trovato alcun riferimento incrociato locale, vengono ricercati anche tutti i tipi di oggetto degli inventari intersphinx.

Riferimenti incrociati ad oggetti

Questi ruoli sono descritti con i rispettivi domini:

- *Python*
- *C*
- *C++*
- *JavaScript*
- *ReST*

Riferimenti incrociati di posizioni arbitrarie

:ref:

Per supportare i riferimenti incrociati a posizioni arbitrarie in qualsiasi documento, vengono utilizzate le etichette [label] reST standard. Affinché ciò funzioni, i nomi delle etichette devono essere univoci in tutta la documentazione. Ci sono due modi per fare riferimento alle etichette:

- Se si inserisce un'etichetta subito prima del titolo di una sezione, vi si può fare riferimento con `:ref:`label-name``. Per esempio:

```
.. _my-reference-label:

Section to cross-reference
-----

This is the text of the section.

It refers to the section itself, see :ref:`my-reference-label`.
```

Il ruolo `:ref:` genererebbe quindi un link alla sezione, avente come titolo "Sezione a cui fare riferimento". Funziona altrettanto bene quando la sezione e il riferimento si trovano in file sorgenti diversi.

Le etichette automatiche funzionano anche con le figure. Per esempio:

```
.. _my-figure:

.. figure:: whatever

   Figure caption
```

In questo caso, un riferimento `:ref:`la-mia-figura`` inserirà un riferimento alla figura col testo del link "Didascalia della figura".

Lo stesso vale per le tabelle a cui viene assegnata una didascalia esplicita utilizzando la direttiva `table`¹⁷⁸.

- È comunque possibile fare riferimento alle etichette che non sono posizionate prima del titolo di una sezione, ma è necessario fornire al link un titolo esplicito, utilizzando questa sintassi: `:ref:`Titolo del Link <label-name>``.

Nota: Le etichette per i riferimenti devono iniziare con un carattere di sottolineatura [underscore]. Quando si fa riferimento a un'etichetta, il carattere di sottolineatura deve essere omesso (vedere gli esempi precedenti).

Si consiglia di utilizzare `ref` rispetto ai link standard reStructuredText alle sezioni (come ``Titolo sezione``) perché funziona su più file, quando i titoli delle sezioni vengono

modificati, genererà dei warning se errato e funziona per tutti i builder che supportano i riferimenti incrociati.

Riferimenti incrociati a documenti

Added in version 0.6.

C'è anche un modo per "linkare" i documenti:

:doc:

Link al documento specificato; il nome del documento può essere specificato in modo assoluto o relativo. Ad esempio, se il riferimento `:doc:`parrot`` capita nel documento `sketches/index`, allora il link si riferisce a `sketches/parrot`. Se il riferimento è `:doc:`/people`` o `:doc:`../people``, il link si riferisce a `people`.

Se non viene fornito alcun testo esplicito per il link (come al solito: `:doc:`Monty Python members </people>``), la didascalia del link sarà il titolo del documento specificato.

Riferimento a file scaricabili

Added in version 0.6.

:download:

Questo ruolo consente di collegarsi ai file all'interno dell'albero dei sorgenti che non sono documenti reST visualizzabili, ma anche a file che possono essere scaricati.

Quando si usa questo ruolo, il file di riferimento viene automaticamente contrassegnato per essere incluso nell'output durante la build (ovviamente, solo per l'output HTML). Tutti i file scaricabili vengono inseriti in una sottodirectory `_downloads/<unique hash>/` della directory di output; vengono gestiti i nomi di duplicati dei file.

Un esempio:

```
See :download:`this example script <../example.py>`.
```

Il nome del file specificato è solitamente relativo alla directory in cui è contenuto il file sorgente corrente, ma se è assoluto (iniziando con `/`), viene considerato relativo alla directory sorgente principale.

Il file `example.py` verrà copiato nella directory di output e verrà generato un link adatto.

Per non visualizzare i link di download non disponibili, si devono racchiudere [wrap] interi paragrafi che hanno questo ruolo:

```
.. only:: builder_html
```

```
See :download:`this example script <../example.py>`.
```

¹⁷⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table>

Riferimenti incrociati a figure tramite il numero

Added in version 1.3.

Cambiato nella versione 1.5: Il ruolo `numref` può anche fare riferimento a sezioni. E `numref` consente `{name}` per il testo del link.

:numref:

Link a figure, tabelle, blocchi di codice e sezioni specificati; vengono utilizzate le etichette standard reST. Quando si utilizza questo ruolo, verrà inserito un riferimento alla figura con il testo del link tramite il numero della figura come "Fig. 1.1".

Se viene fornito un testo esplicito per il link (come al solito: `:numref: Immagine di Sphinx (Fig. %s) <my-figure>`) la didascalia del link fungerà da titolo del riferimento. Come segnaposto, `%s` e `{number}` vengono sostituiti dal numero della figura e `{name}` dalla didascalia della figura. Se non viene fornito alcun testo esplicito del link, l'impostazione `numfig_format` viene utilizzata come default di riserva.

Se `numfig` è `False`, le figure non sono numerate, quindi questo ruolo inserisce non un riferimento ma l'etichetta o il testo del link.

Riferimenti incrociati ad altri elementi

I seguenti ruoli creano eventualmente un riferimento incrociato, ma non si riferiscono a oggetti:

:envvar:

Una variabile d'ambiente. Vengono generate le voci dell'indice. Genera anche un link alla corrispondente direttiva `envvar`, se esiste.

:token:

Il nome di un token grammaticale (utilizzato per creare link tra le direttive `productionlist`).

:keyword:

Il nome di una parola chiave in Python. Ciò crea un link a un'etichetta di riferimento con quel nome, se esiste.

:option:

Un'opzione della riga di comando per un programma eseguibile. Genera un link a una direttiva `option`, se esiste.

Il seguente ruolo crea un riferimento incrociato a un termine in un `glossary`:

:term:

Riferimento a un termine in un glossario. Un glossario viene creato utilizzando la direttiva `glossary` contenente un elenco di definizioni con termini e definizioni. Non deve essere nello stesso file del markup `term`, ad esempio i documenti Python hanno un glossario globale nel file `glossary.rst`.

Se si usa un termine non spiegato in un glossario, si riceverà un warning durante la build.

4.4 Configurazione

La *configuration directory* (directory di configurazione) deve contenere un file chiamato `conf.py`. Questo file (contenente codice Python) è chiamato "file di configurazione della build" e contiene (quasi) tutta la configurazione necessaria per personalizzare il comportamento dell'input e dell'output di Sphinx.

Alla directory di configurazione si può aggiungere un file opzionale `docutils.conf`¹⁷⁹ per regolare la configurazione di `Docutils`¹⁸⁰ se non viene altrimenti sovrascritto o impostato da Sphinx.

Il file di configurazione viene eseguito come codice Python in fase di build (usando `importlib.import_module()`¹⁸¹ e con la directory corrente impostata sulla directory che lo contiene), e quindi può eseguire codice arbitrariamente complesso. Sphinx legge poi nomi semplici dallo spazio dei nomi del file come configurazione.

Punti importanti da notare:

- Se non diversamente documentato, i valori devono essere stringhe e il loro valore di default è la stringa vuota.
- Il termine "fully-qualified name" si riferisce a una stringa che nomina un oggetto Python importabile all'interno di un modulo; ad esempio, il FQN `"sphinx.builders.Builder"` indica la classe `Builder` nel modulo `sphinx.builders`.
- Ricordare che i nomi dei documenti utilizzano `/` come separatore del path e non contengono l'estensione del nome file.
- Poiché `conf.py` viene letto come un file Python, si applicano le solite regole per le codifiche e il supporto Unicode.
- I contenuti dello spazio dei nomi della configurazione vengono serializzati (in modo che Sphinx possa scoprire quando la configurazione cambia), quindi potrebbe non contenere valori non selezionabili: eliminali dallo spazio dei nomi con `del` se è il caso. I moduli vengono rimossi automaticamente, quindi non è necessario un `del` delle importazioni dopo l'uso.
- C'è un oggetto speciale chiamato `tags` disponibile nel file di configurazione. Si può usare per interrogare e modificare i tag (vedere *Inclusione di contenuto basato sui tag*). Utilizzare `'tag'` in `tags` per interrogare, `tags.add('tag')` e `tags.remove('tag')` per modificare. Solo i tag impostati tramite l'opzione della riga di comando `-t` o tramite `tags.add('tag')` si possono interrogare con `'tag'` in `tags`. Si tenga presente che il tag del builder corrente non è disponibile in `conf.py`, poiché viene creato *dopo* che il builder è stato inizializzato.

¹⁷⁹ <https://docutils.sourceforge.io/docs/user/config.html>

¹⁸⁰ <https://docutils.sourceforge.io/>

¹⁸¹ https://docs.python.org/3/library/importlib.html#importlib.import_module

Informazioni sul Progetto

project

Il nome del progetto documentato.

author

Il nome/i dell'autore/i del documento. Il valore di default è 'unknown'.

copyright

Una dichiarazione di copyright secondo lo stile '2008, Nome Autore'.

Cambiato nella versione 7.1: Il valore ora può essere una sequenza di dichiarazioni di copyright nella forma precedente, che verranno visualizzate ciascuna nella propria riga.

project_copyright

Un alias di *copyright*.

Added in version 3.5.

version

La versione 'major' del progetto, utilizzata in sostituzione di `|version|`. Per esempio, per la documentazione Python, potrebbe essere qualcosa come 2.6.

release

La versione completa del progetto, utilizzata in sostituzione di `|release|` e ad es. nei template HTML. Per esempio, per la documentazione Python, potrebbe essere qualcosa come 2.6.0rc1.

Se non c'è bisogno della separazione fornita tra *version* e *release*, si settano semplicemente con lo stesso valore.

Configurazione generale

extensions

Un elenco di stringhe che rappresentano i nomi dei moduli delle estensioni *extensions*. Possono essere estensioni fornite con Sphinx (chiamate `sphinx.ext.*`) o quelle custom.

Si tenga presente che si può estendere `sys.path`¹⁸² all'interno del file conf se le estensioni risiedono in un'altra directory, ma si devono utilizzare path assoluti. Se il path dell'estensione è relativo alla *configuration directory*, si usa `os.path.abspath()`¹⁸³ in questo modo:

```
import sys, os

sys.path.append(os.path.abspath('sphinxext'))

extensions = ['extname']
```

In questo modo, si può caricare un'estensione chiamata `extname` dalla subdirectory `sphinxext`.

Il file di configurazione stesso può essere un'estensione; per questo, basta fornire una funzione `setup()` al suo interno.

source_suffix

I file di estensione dei file sorgenti. Sphinx considera i file con questo suffisso come sorgenti. Il valore può essere un dizionario [dictionary] che associa le estensioni ai tipi di file. Per esempio:

```
source_suffix = {
    '.rst': 'restructuredtext',
    '.txt': 'restructuredtext',
    '.md': 'markdown',
}
```

Per default, Sphinx supporta solo il tipo di file `'restructuredtext'`. Si può aggiungere un nuovo tipo di file utilizzando le estensioni del parser dei sorgenti. Si legga un documento dell'estensione per sapere quale tipo di file supporta l'estensione.

Il valore può anche essere un elenco di estensioni di file: in tal caso Sphinx considererà che siano tutte mappate al tipo di file `'restructuredtext'`

Il default è `{'.rst': 'restructuredtext'}`.

Nota: le estensioni dei file devono iniziare con un punto (es. `.rst`).

Cambiato nella versione 1.3: Ora può essere un elenco di estensioni.

Cambiato nella versione 1.8: Supporto della mappatura del tipo di file

source_encoding

La codifica di tutti i file sorgente reST. La codifica consigliata e il valore di default è `'utf-8-sig'`.

Added in version 0.5: In precedenza, Sphinx accettava solo sorgenti con codifica UTF-8.

source_parsers

Se fornito, è sufficiente un dizionario di classi parser per diversi sorgenti. Le chiavi sono il suffisso, i valori possono essere una classe o una stringa che fornisce un nome completo di una classe parser. La classe parser può essere `docutils.parsers.Parser` oppure `sphinx.parsers.Parser`. I file con un suffisso non presente nel dizionario verranno analizzati con il parser `reStructuredText` di default.

Per esempio:

```
source_parsers = {'.md': 'recommonmark.parser.CommonMarkParser'}
```

¹⁸² <https://docs.python.org/3/library/sys.html#sys.path>

¹⁸³ <https://docs.python.org/3/library/os.path.html#os.path.abspath>

Nota: Fare riferimento [Markdown](#) per ulteriori informazioni sull'utilizzo di Markdown con Sphinx.

Added in version 1.3.

Deprecato dalla versione 1.8: Ora Sphinx fornisce un'API [Sphinx.add_source_parser\(\)](#) per registrare un parser sorgente. Si prega di usarlo in sostituzione.

master_doc

Uguale a [root_doc](#).

Cambiato nella versione 4.0: Rinominato master_doc in root_doc.

root_doc

Il nome del documento "root", cioè il documento che contiene la direttiva root [toctree](#). Il default è 'index'.

Cambiato nella versione 2.0: Il default è cambiato in 'index' era 'contents'.

Cambiato nella versione 4.0: Rinominato root_doc era master_doc.

exclude_patterns

Un elenco di pattern in stile glob¹ che dovrebbero essere esclusi durante la ricerca di file sorgenti. Vengono confrontati con i nomi dei sorgenti relativi alla directory dei sorgenti, utilizzando le barre come separatori di directory su tutte le piattaforme.

Pattern di esempio:

- 'library/xml.rst' -- ignora il file library/xml.rst
- 'library/xml' -- ignora la directory library/xml
- 'library/xml*' -- ignora tutti i file e le directory che iniziano con library/xml
- '**/.svn' -- ignora tutte le directory .svn

[exclude_patterns](#) viene consultato anche quando si cercano file statici in [html_static_path](#) e in [html_extra_path](#).

Added in version 1.0.

include_patterns

Un elenco di pattern in stile glob¹ utilizzati per trovare i file sorgenti. Vengono confrontati con i nomi dei sorgenti relativi alla directory dei sorgenti, utilizzando le barre come separatori di directory su tutte le piattaforme. Il default è **, il che significa che tutti i file vengono inclusi ricorsivamente dalla directory dei sorgenti. [exclude_patterns](#) ha la priorità su [include_patterns](#).

Pattern di esempio:

¹ Una nota sulla sintassi del globbing: si possono usare i costrutti standard della shell *, ?, [...] e [!...] con la caratteristica che tutti questi non corrispondono alle barre [slash]. Si può usare un doppio asterisco ** per corrispondere a qualsiasi sequenza di caratteri *inclusi* gli slash.

- `'**'` -- tutti i file nella directory dei sorgenti e nelle sottodirectory, ricorsivamente
- `'library/xml'` -- solo la directory `library/xml`
- `'library/xml*'` -- tutti i file e le directory che iniziano con `library/xml`
- `'**/doc'` -- tutte le directory `doc` (questo potrebbe essere utile se la documentazione è co-localizzata con i file sorgenti)

Added in version 5.1.

templates_path

Un elenco di path che contengono template aggiuntivi (o template che sovrascrivono quelli nativi/specifici del tema). I path relativi vengono considerati relativi alla directory di configurazione.

Cambiato nella versione 1.3: Dato che questi file non sono pensati per essere compilati, vengono automaticamente aggiunti a [*exclude_patterns*](#).

template_bridge

Una stringa con il nome completo di un 'callable' (o semplicemente di una classe) che restituisce un'istanza di [*TemplateBridge*](#). Questa istanza viene poi utilizzata per eseguire il rendering dei documenti HTML e possibilmente dell'output di altri builder (attualmente il builder delle modifiche). (Si noti che il bridge del template deve essere reso sensibile al tema se si devono utilizzare temi HTML).

rst_epilog

Una stringa di reStructuredText che verrà inclusa alla fine di ogni file sorgente letto. Questo è un posto possibile per aggiungere sostituzioni che dovrebbero essere disponibili in ogni file (un altro è [*rst_prolog*](#)). Un esempio:

```
rst_epilog = """
.. |psf| replace:: Python Software Foundation
"""
```

Added in version 0.6.

rst_prolog

Una stringa di reStructuredText che verrà inclusa all'inizio di ogni file sorgente letto. Questo è un posto possibile per aggiungere sostituzioni che dovrebbero essere disponibili in ogni file (un altro è [*rst_epilog*](#)). Un esempio:

```
rst_prolog = """
.. |psf| replace:: Python Software Foundation
"""
```

Added in version 1.0.

primary_domain

Il nome del [*domain*](#) di default. Può anche essere `None` per disabilitare un dominio di default. Il default è `'py'`. Quegli oggetti in altri domini (sia che il nome di dominio sia dato esplicitamente, o selezionato da una direttiva [*default-domain*](#)) avranno il nome del dominio esplicitamente anteposto quando nominati (ad esempio, quando il nome del

dominio di default è C, le funzioni Python verranno chiamate "funzione Python", non solamente "funzione").

Added in version 1.0.

default_role

Il nome di un ruolo reST (nativo o estensione Sphinx) da utilizzare come ruolo di default, ovvero per il testo contrassegnato ``come questo``. Questo può essere impostato su `'py:obj'` per fare di ``filter`` un riferimento incrociato alla funzione Python "filter". Il default è None, che non riassegna il ruolo di default.

Il ruolo di default può sempre essere impostato all'interno di singoli documenti utilizzando la direttiva standard reST `default-role`¹⁸⁴.

Added in version 0.4.

keep_warnings

Se 'true', mantiene i warning come paragrafi di "messaggi di sistema" nei documenti creati. Indipendentemente da questa impostazione, i warning vengono sempre scritti nel flusso degli errori standard quando viene eseguito `sphinx-build`.

Il default è False, il comportamento pre-0.5 era di mantenerli sempre.

Added in version 0.5.

show_warning_types

Se True, il tipo di ciascun warning viene aggiunto come suffisso al messaggio di warning, ad esempio, `WARNING: [...] [index]` o `WARNING: [...] [toc.circular]`. Il default è False.

Added in version 7.3.0.

suppress_warnings

Un elenco di tipi di warning per eliminare quelli arbitrari.

Il core Sphinx supporta i seguenti tipi di warning:

- `app.add_node`
- `app.add_directive`
- `app.add_role`
- `app.add_generic_role`
- `app.add_source_parser`
- `config.cache`
- `download.not_readable`
- `epub.unknown_project_files`
- `epub.duplicated_toc_entry`
- `i18n.inconsistent_references`

¹⁸⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#default-role>

- `index`
- `image.not_readable`
- `ref.term`
- `ref.ref`
- `ref.numref`
- `ref.keyword`
- `ref.option`
- `ref.citation`
- `ref.footnote`
- `ref.doc`
- `ref.python`
- `misc.highlighting_failure`
- `toc.circular`
- `toc.excluded`
- `toc.not_readable`
- `toc.secnum`
- `toc.no_title`

Le estensioni possono anche definire i propri tipi di warning. Quelli definiti dalle estensioni proprietarie `sphinx.ext` sono:

- `autodoc`
- `autodoc.import_object`
- `autosectionlabel.<document name>`
- `autosummary`
- `intersphinx.external`

Si può scegliere tra questi tipi. Si può anche fornire solo il primo componente per escludere tutti i warning ad esso allegati.

Added in version 1.4.

Cambiato nella versione 1.5: Added `misc.highlighting_failure`

Cambiato nella versione 1.5.1: Aggiunto `epub.unknown_project_files`

Cambiato nella versione 1.6: Aggiunto `ref.footnote`

Cambiato nella versione 2.1: Aggiunto `autosectionlabel.<document name>`

Cambiato nella versione 3.3.0: Aggiunto `epub.duplicated_toc_entry`

Cambiato nella versione 4.3: Aggiunti `toc.excluded` e `toc.not_readable`

Added in version 4.5: Aggiunto `inconsistent_references`

Added in version 7.1: Aggiunto il tipo di warning `index`.

Added in version 7.3: Aggiunto il tipo di warning `config.cache`.

needs_sphinx

Se impostato su una stringa di versione `major.minor` come `'1.1'`, Sphinx lo confronterà con la sua versione e rifiuterà di compilare se è troppo vecchio. Il default non è richiesto.

Added in version 1.0.

Cambiato nella versione 1.4: accetta anche la stringa della versione `micro`

needs_extensions

Questo valore può essere un dizionario che specifica i requisiti di versione per le estensioni in *extensions*, ad es. `needs_extensions = {'sphinxcontrib.something': '1.5'}`. Le stringhe della versione dovrebbero essere nel formato `major.minor`. Non è necessario specificare i requisiti per tutte le estensioni, ma solo per quelle che si desiderano verificare.

Questo richiede che l'estensione specifichi la sua versione di Sphinx (vedere *Le API di Sphinx* per come farlo).

Added in version 1.3.

manpages_url

Un URL per fare riferimenti incrociati ai ruoli *manpage*. Se questo è definito su `https://manpages.debian.org/{path}`, il ruolo `:manpage: `man(1)`` si collegherà a `<https://manpages.debian.org/man(1)>`. I pattern disponibili sono:

- `page` - la pagina del manuale (`man`)
- `section` - la sezione del manuale (`1`)
- `path` - la pagina del manuale originale e la sezione specificata (`man(1)`)

Questo supporta anche le pagine `man` specificate come `man.1`.

Nota: Attualmente interessa solo gli autori HTML ma potrebbe essere ampliato in futuro.

Added in version 1.7.

nitpicky

Se `true`, Sphinx emetterà un warning su *tutti* i riferimenti in cui non è possibile trovare il target. Il default è `False`. Si può attivare temporaneamente questa modalità utilizzando l'opzione `-n` della riga di comando.

Added in version 1.0.

nitpick_ignore

Un 'set' o un elenco di tuple (`type`, `target`) (per default vuoto) che dovrebbero essere

ignorare quando generano warning in "nitpicky mode". Si noti che `type` dovrebbe includere il nome del dominio, se presente. Voci di esempio potrebbero essere `('py:func', 'int')` o `('envvar', 'LD_LIBRARY_PATH')`.

Added in version 1.1.

Cambiato nella versione 6.2: Modificati i tipi di contenitori consentiti in un set, un elenco o una tupla

nitpick_ignore_regex

Una versione estesa di [`nitpick_ignore`](#), che interpreta invece le stringhe `type` e `target` come espressioni regolari. Notare che l'espressione regolare deve corrispondere all'intera stringa (come se fossero inseriti i marcatori `^` e `$`).

Per esempio, `(r'py:.*', r'foo.*bar\.B.*')` ignorerà i warning 'nitpicky' per tutte le entità Python che iniziano con 'foo' e contengono 'bar.B', come `('py:const', 'foo_package.bar.BAZ_VALUE')` o `('py:class', 'food.bar.Barman')`.

Added in version 4.1.

Cambiato nella versione 6.2: Modificati i tipi di contenitori consentiti in un set, un elenco o una tupla

numfig

Se `true`, figure, tabelle e blocchi di codice vengono automaticamente numerati se hanno una didascalia. Il ruolo [`numref`](#) è abilitato. Finora obbedisce solo ai builder HTML e LaTeX. Il default è `False`.

Nota: Il builder LaTeX assegna sempre i numeri indipendentemente dal fatto che questa opzione sia abilitata

Added in version 1.3.

numfig_format

Un dizionario che mappa 'figure', 'table', 'code-block' e 'section' a stringhe utilizzate per il formato dei numeri delle figure. Come carattere speciale, `%s` verrà sostituito dal numero della figura.

Il default è quello di usare 'Fig. %s' per 'figure', 'Table %s' per 'table', 'Listing %s' per 'code-block' e 'Section %s' per 'section'.

Added in version 1.3.

numfig_secnum_depth

- se impostato a `0`, figure, tabelle e blocchi di codice vengono numerati in modo continuo a partire da 1.
- se 1 (il default) i numeri saranno `x.1`, `x.2`, ... con `x` il numero di sezione (di livello superiore; nessun `x`. se non ci sono sezioni). Ciò si applica naturalmente solo se la numerazione delle sezioni è stata attivata tramite l'opzione `:numbered:` della direttiva [`toctree`](#).

- 2 significa che i numeri saranno `x.y.1`, `x.y.2`, ... se posizionati in una sottosezione (ma comunque `x.1`, `x.2`, ... se si trovano direttamente sotto una sezione e 1, 2, ... se non c'è nessuna sezione di livello superiore).
- ecc...

Added in version 1.3.

Cambiato nella versione 1.7: Il builder LaTeX obbedisce a questa impostazione (se `numfig` è settato a `True`).

smartquotes

Se `true`, la [Docutils Smart Quotes transform](#)¹⁸⁵, originariamente basata su [SmartyPants](#)¹⁸⁶ (limitata all'inglese) e attualmente applicata a molte lingue, verrà utilizzata per convertire virgolette e trattini nelle entità tipograficamente corrette. Il default è: `True`.

Added in version 1.6.6: Sostituisce il deprecato `html_use_smartypants`. Si applica per default a tutti i builder tranne `man` e `text` (vedere [smartquotes_excludes](#)).

Un file `docutils.conf`¹⁸⁷ situato nella directory di configurazione (o un file `~/.docutils` globale) viene rispettato incondizionatamente se *disattiva* le virgolette smart tramite la corrispondente [opzione Docutils](#)¹⁸⁸. Ma se li *attiva*, allora prevale `smartquotes`.

smartquotes_action

Questa stringa personalizza la trasformazione delle "Smart Quotes". Consultare il file `smartquotes.py` nel [Docutils repository](#)¹⁸⁹ per i dettagli. Il default `'qDe'` istruisce i normali caratteri `quote` (virgolette) `"`, `'`, `em-` e `en-`Dashes (trattini) `---`, `--`, e ellipses `...` (punti sospensivi).

Added in version 1.6.6.

smartquotes_excludes

Questo è un dict il cui default è:

```
{'languages': ['ja'], 'builders': ['man', 'text']}
```

Ogni voce fornisce una condizione sufficiente per ignorare l'impostazione `smartquotes` e disattivare la trasformazione delle Virgolette Smart. Le chiavi accettate sono quelle sopra `'builders'` o `'languages'`. I valori sono liste.

Nota: Attualmente, in caso di invocazione di `make` con più target, il nome del primo target è l'unico che viene testato rispetto alla voce `'builders'` e decide per tutti. Inoltre, un `make text` successivo a `make html` deve essere emesso nella forma `make text 0="-E"` per forzare la ri-analisi dei file sorgente, poiché quelli memorizzati nella cache sono già trasformati. D'altra parte il problema non si pone con l'uso diretto di

¹⁸⁵ <https://docutils.sourceforge.io/docs/user/smartquotes.html>

¹⁸⁶ <https://daringfireball.net/projects/smartypants/>

¹⁸⁷ <https://docutils.sourceforge.io/docs/user/config.html>

¹⁸⁸ <https://docutils.sourceforge.io/docs/user/config.html#smart-quotes>

¹⁸⁹ <https://sourceforge.net/p/docutils/code/HEAD/tree/trunk/docutils/>

sphinx-build poiché memorizza nella cache (per default) i file sorgente analizzati nelle posizioni per ogni builder.

Suggerimento: Un modo alternativo per disattivare (o personalizzare) in modo efficace le virgolette smart per un determinato builder, ad esempio `latex`, è utilizzare `make` in questo modo:

```
make latex O="-D smartquotes_action="
```

Ciò può seguire alcuni `make html` senza problemi, contrariamente alla situazione della nota precedente.

Added in version 1.6.6.

user_agent

Un User-Agent di Sphinx. Viene usato per un header sull'accesso HTTP (es. `linkcheck`, `intersphinx` e così via). Il default è `"Sphinx/X.Y.Z requests/X.Y.Z python/X.Y.Z"`.

Added in version 2.3.

tls_verify

Se `true`, Sphinx verifica le certificazioni del server. Il default è `True`.

Added in version 1.5.

tls_cacerts

Un path a un file di certificazione della CA o uno alla directory che contiene i certificati. Ciò consente inoltre a un dizionario di mappare il nome host al path del file del certificato. I certificati vengono utilizzati per verificare i server.

Added in version 1.5.

Suggerimento: Sphinx usa internamente `requests`¹⁹⁰ come libreria HTTP. Pertanto, Sphinx fa riferimento a un file di certificazione nella directory puntata dalla variabile di ambiente `REQUESTS_CA_BUNDLE` se `tls_cacerts` non è settato.

today

today_fmt

Questi valori determinano come formattare la data corrente, utilizzata come sostituzione di `|today|`.

- Se si setta `today` su un valore non vuoto, verrà utilizzato.
- Altrimenti, l'ora corrente viene formattata utilizzando `time.strftime()`¹⁹¹ e il formato fornito in `today_fmt`.

¹⁹⁰ <https://requests.readthedocs.io/en/master/>

Il default è ora *today* e un *today_fmt* di '%b %d, %Y' (o, se la traduzione è abilitata con *language*, un formato equivalente per la lingua selezionata).

highlight_language

Il linguaggio di default in cui evidenziare il codice sorgente. Il default è 'default'. È simile a 'python3'; è principalmente un superset di 'python' ma ritorna a 'none' senza alcun warning in caso di errore. 'python3' e altri linguaggi emetteranno un warning in caso di errore.

Il valore deve essere un nome lexer Pygments valido, vedere *Mostrare esempi di codice* per maggiori dettagli.

Added in version 0.5.

Cambiato nella versione 1.4: Ora il default è 'default'. Se si preferisce evidenziare solamente Python 2, lo si può settare a 'python'.

highlight_options

Un dictionary che associa i nomi dei linguaggi alle opzioni per i moduli lexer di Pygments. Sono specifici del lexer; per le opzioni di ciascuno, consultare la *documentazione di Pygments*¹⁹².

Esempio:

```
highlight_options = {
    'default': {'stripall': True},
    'php': {'startinline': True},
}
```

È consentito anche un dizionario unico delle opzioni. Quindi viene riconosciuto come opzione per il lexer specificato da *highlight_language*:

```
# configuration for the `highlight_language`
highlight_options = {'stripall': True}
```

Added in version 1.3.

Cambiato nella versione 3.5: Consente di configurare le opzioni di evidenziazione per più lingueaggi

pygments_style

Il nome dello stile da utilizzare per l'evidenziazione di Pygments del codice sorgente. Se non impostato, per l'output HTML verrà selezionato lo stile di default del tema o 'sphinx'.

Cambiato nella versione 0.3: Se il valore è un nome completo di una classe di uno stile Pygments personalizzata, viene utilizzato come stile personalizzato.

maximum_signature_line_length

Se la lunghezza in caratteri di una firma [signature] supera il numero impostato, ciascun parametro all'interno della firma verrà visualizzato su una singola riga logica.

¹⁹¹ <https://docs.python.org/3/library/time.html#time.strftime>

¹⁹² <https://pygments.org/docs/lexers>

Quando è `None` (il default), non esiste una lunghezza massima e l'intera firma verrà visualizzata su un'unica riga logica.

Una 'riga logica' è simile a un'andata a capo: i builder o i temi possono scegliere di una 'soft wrap' (andata a capo) di una singola riga logica e questa impostazione non influisce su tale comportamento.

I domini possono fornire opzioni per eliminare qualsiasi andata a capo su una direttiva di un singolo oggetto, come visto nei domini C, C++ e Python (ad esempio [py:function:single-line-parameter-list](#)).

Added in version 7.1.

add_function_parentheses

Un booleano che decide se le parentesi vengono aggiunte al testo del ruolo del metodo o della funzione (ad esempio il contenuto di `:func:`input``) per indicare che il nome è richiamabile. Il default è `True`.

add_module_names

Un booleano che decide se i nomi dei moduli sono anteposti a tutti i nomi *object* (per i tipi di oggetto in cui è definito un "modulo" di qualche tipo), ad es. per le direttive [py:function](#). Il default è `True`.

toc_object_entries

Crea voci di sommario per oggetti del dominio (ad esempio funzioni, classi, attributi, ecc.). Il default è `True`.

toc_object_entries_show_parents

Una stringa che determina come gli oggetti del dominio (ad esempio funzioni, classi, attributi, ecc.) vengono visualizzati nella voce del sommario.

Usa `domain` per consentire al dominio di determinare il numero appropriato di genitori da mostrare. Ad esempio, il dominio Python mostrerebbe `Class.method()` e `function()`, tralasciando il livello `module.` dei genitori. Questa è l'impostazione di default.

Usa `hide` per mostrare solo il nome dell'elemento senza genitori (ad esempio `method()`).

Usa `all` per mostrare il nome completo dell'oggetto (ad esempio `module.Class.method()`), mostrando tutti i genitori.

Added in version 5.2.

show_authors

Un valore booleano che decide se le direttive [codeauthor](#) e [sectionauthor](#) producono qualsiasi output nei file costruiti.

modindex_common_prefix

Un elenco di prefissi che vengono ignorati per l'ordinamento dell'indice del modulo Python (ad esempio, se questo è impostato su `['foo. ']`, allora `foo.bar` viene mostrato sotto B, non F). Questo può essere utile se si documenta un progetto costituito da un singolo pacchetto. Funziona solo per il builder HTML al momento. Il default è `[]`.

Added in version 0.6.

trim_footnote_reference_space

Rimuove gli spazi prima dei riferimenti alle note a piè di pagina necessari affinché il parser reST riconosca la nota a piè di pagina, ma non appaiono gradevoli nell'output.

Added in version 0.6.

trim_doctest_flags

Se "true", i flag doctest (commenti simili a `# doctest: FLAG, ...`) alla fine delle righe e dei marcatori `<BLANKLINE>` vengono rimossi per tutti i blocchi di codice che mostrano interattività tra le sessioni Python (cioè doctests). Il default è True. Vedere l'estensione [doctest](#) per ulteriori possibilità di includere doctest.

Added in version 1.0.

Cambiato nella versione 1.1: Ora rimuove anche `<BLANKLINE>`.

strip_signature_backslash

Il default è False. Quando è abilitata la rimozione col backslash, ogni occorrenza di `\\` in una direttiva di dominio verrà modificata in `\`, anche all'interno delle stringhe letterali. Questo era il comportamento prima della versione 3.0 e l'impostazione di questa variabile su True lo ripristinerà.

Added in version 3.0.

option_emphasise_placeholders

Il default è False. Se abilitato, enfatizza i segnaposto [placeholder] nelle direttive [option](#). Per visualizzare le parentesi graffe letterali, utilizzare una barra rovesciata [backslash] (`\{`). Per esempio, `option_emphasise_placeholders=True` e `.. option:: -foption={TYPE}` verrebbe visualizzato con TYPE enfatizzato.

Added in version 5.1.

Opzioni per l'internazionalizzazione

Queste opzioni influenzano il *Native Language Support* di Sphinx. Consultare la documentazione su [Internazionalizzazione](#) per i dettagli.

language

Il codice per la lingua in cui sono scritti i documenti. Qualsiasi testo generato automaticamente da Sphinx sarà in tale linguaggio. Inoltre, Sphinx proverà a sostituire singoli paragrafi dei documenti con i set di traduzione ottenuti da [locale_dirs](#). Sphinx cercherà le figure specifiche della lingua denominate da [figure_language_filename](#) (ad esempio, la versione tedesca di `myfigure.png` sarà `myfigure.de.png` per default) e sostituirli con le figure originali. Nel builder LaTeX, verrà selezionata una lingua adatta come opzione per il pacchetto *Babel*. Il default è 'en'.

Added in version 0.5.

Cambiato nella versione 1.4: Supporta la sostituzione delle figure

Cambiato nella versione 5.0.

Le lingue attualmente supportate da Sphinx sono:

- ar -- Arabic
- bg -- Bulgarian
- bn -- Bengali
- ca -- Catalan
- cak -- Kaqchikel
- cs -- Czech
- cy -- Welsh
- da -- Danish
- de -- German
- el -- Greek
- en -- English (default)
- eo -- Esperanto
- es -- Spanish
- et -- Estonian
- eu -- Basque
- fa -- Iranian
- fi -- Finnish
- fr -- French
- he -- Hebrew
- hi -- Hindi
- hi_IN -- Hindi (India)
- hr -- Croatian
- hu -- Hungarian
- id -- Indonesian
- it -- Italian
- ja -- Japanese
- ko -- Korean
- lt -- Lithuanian
- lv -- Latvian
- mk -- Macedonian
- nb_NO -- Norwegian Bokmal
- ne -- Nepali

- `nl` -- Dutch
- `pl` -- Polish
- `pt` -- Portuguese
- `pt_BR` -- Brazilian Portuguese
- `pt_PT` -- European Portuguese
- `ro` -- Romanian
- `ru` -- Russian
- `si` -- Sinhala
- `sk` -- Slovak
- `sl` -- Slovenian
- `sq` -- Albanian
- `sr` -- Serbian
- `sr@latin` -- Serbian (Latin)
- `sr_RS` -- Serbian (Cyrillic)
- `sv` -- Swedish
- `ta` -- Tamil
- `te` -- Telugu
- `tr` -- Turkish
- `uk_UA` -- Ukrainian
- `ur` -- Urdu
- `vi` -- Vietnamese
- `zh_CN` -- Simplified Chinese
- `zh_TW` -- Traditional Chinese

locale_dirs

Added in version 0.5.

Le directory in cui cercare ulteriori cataloghi di messaggi (vedere [*language*](#)), relative alla directory dei sorgenti. Le directory su questo path vengono cercate dal modulo standard [*gettext*](#)¹⁹³.

I messaggi interni vengono recuperati da un dominio text di `sphinx`; quindi se si aggiunge la directory `./locale` a questa impostazione, i cataloghi dei messaggi (compilati dal formato `.po` utilizzando `msgfmt`) devono stare in `./locale/language/LC_MESSAGES/sphinx.mo`. Il dominio text dei singoli documenti dipende da [*gettext_compact*](#).

Il default è `['locales']`.

Nota: L'opzione *-v option for sphinx-build command* è utile per verificare che la configurazione di `locale_dirs` funzioni come previsto. Emette messaggi di debug se la directory del catalogo dei messaggi non viene trovata.

Cambiato nella versione 1.5: Utilizza la directory `locales` come valore di default

gettext_allow_fuzzy_translations

Se true, per la traduzione vengono utilizzati i messaggi "fuzzy" nei cataloghi dei messaggi. Il default è False.

Added in version 4.3.

gettext_compact

Added in version 1.1.

Se true, il dominio text di un documento è il suo `namedoc` se si tratta di un file di progetto di livello superiore, altrimenti la sua directory di base.

Se impostato su stringa, il dominio text di tutto il documento sarà questa stringa, facendo in modo che tutti i documenti utilizzino un singolo dominio text.

Per default, il documento `markup/code.rst` finisce nel dominio text `markup`. Con questa opzione settata su False, è `markup/code`.

Cambiato nella versione 3.3: Il valore stringa è ora accettato.

gettext_uuid

Se true, Sphinx genera informazioni uuid per il rilevamento della versione nei cataloghi dei messaggi. È usato per:

- Aggiungere la riga uid per ogni msgid nei file .pot.
- Calcolare la similarità tra i nuovi msgid e i vecchi msgid precedentemente salvati. Questo calcolo richiede molto tempo.

Per accelerare il calcolo, si può usare `python-levenshtein`, un pacchetto di terze parti scritto in C utilizzando **`pip install python-levenshtein`**.

Il default è False.

Added in version 1.3.

gettext_location

Se true, Sphinx informazioni sulla posizione per i messaggi nei cataloghi di messaggi.

Il default è True.

Added in version 1.3.

gettext_auto_build

Se true, Sphinx crea il file `mo` per ogni file del catalogo di traduzione.

Il default è True.

¹⁹³ <https://docs.python.org/3/library/gettext.html#module-gettext>

Added in version 1.3.

gettext_additional_targets

Per specificare i nomi per abilitare l'estrazione e la traduzione del gettext applicando inoltre i18n. È possibile specificare i nomi seguenti:

Index

termini dell'indice

Literal-block

blocchi letterali (annotazione `::` e direttiva `code-block`)

Doctest-block

blocco doctest

Raw

contenuto grezzo

Image

uri di un'immagine/figura

Per esempio: `gettext_additional_targets = ['literal-block', 'image']`.

Il default è `[]`.

Added in version 1.3.

Cambiato nella versione 4.0: Il testo alternativo dell'immagine è tradotto per default.

figure_language_filename

Il formato del nome file per le figure specifiche della lingua. Il valore di default è `{root}. {language}{ext}`. Verrà espanso in `dirname/filename.en.png` da `.. image:: dirname/filename.png`. I token di formato disponibili sono:

- `{root}` - il nome del file, incluso qualsiasi componente del path, senza l'estensione del file, ad es. `dirname/filename`
- `{path}` - il componente del path della directory del nome del file, con una barra finale se non vuoto, ad es. `dirname/`
- `{docpath}` - il componente del path della directory per il documento corrente, con una barra finale se non vuoto.
- `{basename}` - il nome del file senza il path della directory o i componenti dell'estensione del file, ad es. `filename`
- `{ext}` - l'estensione del file, ad es. `.png`
- `{language}` - la lingua di traduzione, ad es. `en`

Ad esempio, impostandolo su `{path}{language}/{basename}{ext}` si espanderà invece in `dirname/en/filename.png`.

Added in version 1.4.

Cambiato nella versione 1.5: Aggiunti i token `{path}` e `{basename}`.

Cambiato nella versione 3.2: Aggiunto il token `{docpath}`.

translation_progress_classes

Controlla quali classi vengono eventualmente aggiunte per indicare l'avanzamento della traduzione. Questa impostazione verrebbe probabilmente utilizzata solo dai traduttori della documentazione, per indicare rapidamente i contenuti tradotti e non tradotti.

- **True**: aggiunge le classi `translated` e `untranslated` a tutti i nodi con contenuto traducibile.
- **translated**: aggiunge solo la classe `translated`.
- **untranslated**: aggiunge solo la classe `untranslated`.
- **False**: non aggiunge alcuna classe per indicare l'avanzamento della traduzione.

Il default è `False`.

Added in version 7.1.

Opzioni per le Formule

Queste opzioni influenzano le notazioni matematiche.

math_number_all

Impostare questa opzione su `True` se si vuole che tutte le formule visualizzate siano numerate. Il default è `False`.

math_eqref_format

Una stringa utilizzata per formattare le etichette dei riferimenti alle equazioni. Il segnaposto `{number}` indica il numero dell'equazione.

Esempio: `'Eq. {number}'` viene visualizzato come, ad esempio, `Eq. 10`.

math_numfig

Se `True`, le equazioni matematiche visualizzate vengono numerate sulle pagine quando `numfig` è abilitato. L'impostazione `numfig_secnum_depth` viene rispettata. Dev'essere utilizzato il ruolo `eq`, non `numref`, per fare riferimento ai numeri delle equazioni. Il default è `True`.

Added in version 1.7.

Opzioni per l'output HTML

Queste opzioni influenzano l'HTML, l'output dell'Help HTML e altri builder che utilizzano la classe `HTMLWriter` di Sphinx.

html_theme

Il "tema" che dovrebbe utilizzare l'output HTML. Vedere la [sezione sui temi](#). Il default è `'alabaster'`.

Added in version 0.6.

html_theme_options

Un dictionary di opzioni che influenzano l'aspetto del tema selezionato. Questi sono specifici di ciascun tema. Per le opzioni dei temi nativi, consultare [questa sezione](#).

Added in version 0.6.

html_theme_path

Un elenco di path che contengono temi personalizzati, come sottodirectory o come file zip. I path relativi vengono considerati relativi alla directory di configurazione.

Added in version 0.6.

html_style

Il foglio di stile da utilizzare per le pagine HTML. Un file con quel nome deve esistere nel path `static/` di Sphinx, o in uno dei percorsi personalizzati forniti in [html_static_path](#). Il default è il foglio di stile fornito dal tema selezionato. Per aggiungere o sovrascrivere solo alcune cose rispetto al foglio di stile del tema, si usa l'`@import` del CSS per importare il foglio di stile del tema.

html_title

Il "titolo" per la documentazione HTML generata con i template propri di Sphinx. Questo viene aggiunto al tag `<title>` delle singole pagine e utilizzato nella barra di navigazione come elemento "più in alto". Il default è '`<project> v<revision> documentation`'.

html_short_title

Un "titolo" più breve per i documenti HTML. Viene utilizzato per i link nel titolo e nei documenti dell'Help HTML. Se non fornito, il valore di default è [html_title](#).

Added in version 0.4.

html_baseurl

L'URL di base che punta alla radice della documentazione HTML. Serve per indicare la posizione del documento utilizzando [The Canonical Link Relation](#)¹⁹⁴. Il default è: ''.

Added in version 1.8.

html_codeblock_linenos_style

Lo stile dei numeri di riga per i blocchi di codice.

- 'table' -- visualizza i numeri di riga utilizzando il tag `<table>`
- 'inline' -- visualizza i numeri di riga utilizzando il tag `` (il default)

Added in version 3.2.

Cambiato nella versione 4.0: Il default è 'inline'.

Deprecato dalla versione 4.0.

html_context

Un dizionario di valori da passare nel contesto del motore del template per tutte le pagine.

¹⁹⁴ <https://datatracker.ietf.org/doc/html/rfc6596>

In questo dizionario si possono inserire anche singoli valori, utilizzando l'opzione `-A` della riga di comando `sphinx-build`.

Added in version 0.5.

html_logo

Se fornito, deve essere il nome di un'immagine (path relativo alla *configuration directory*) che è il logo dei documenti o un URL che punta a un'immagine per il logo. Viene posizionata nella parte superiore della barra laterale; la sua larghezza non dovrà quindi superare i 200 pixel. Il default è: `None`.

Added in version 0.4.1: L'immagine verrà copiata nella directory `_static` dell'HTML di output, ma solo se il file non è già presente.

Cambiato nella versione 4.0: Accetta anche l'URL per il file del logo.

html_favicon

Se fornito, deve essere il nome di un'immagine (path relativo alla *configuration directory*) che è la favicon dei documenti o un URL che punta a un'immagine per la favicon. I browser moderni lo utilizzano come icona per schede, finestre e segnalibri. Dovrebbe essere un file icona in stile Windows (`.ico`), di dimensioni 16x16 o 32x32 pixel. Il default è: `None`.

Added in version 0.4: L'immagine verrà copiata nella directory `_static` dell'HTML di output, ma solo se il file non è già presente.

Cambiato nella versione 4.0: Accetta anche l'URL per la favicon.

html_css_files

Un elenco di file CSS. La voce deve essere una stringa *filename* o una tupla contenente la stringa *filename* e il dizionario *attributes*. Il *filename* deve essere relativo a *html_static_path*, o un URI completo con uno schema come `https://example.org/style.css`. *attributes* viene usato per gli attributi del tag `<link>`. Il default è una lista vuota.

Esempio:

```
html_css_files = ['custom.css',
                  'https://example.com/css/custom.css',
                  ('print.css', {'media': 'print'})]
```

Come attributo speciale, *priority* può essere impostato come numero intero per caricare il file CSS in un passaggio precedente o più lento. Per ulteriori informazioni, fare riferimento a *Sphinx.add_css_file()*.

Added in version 1.8.

Cambiato nella versione 3.5: Supporto per l'attributo di priorità

html_js_files

Un elenco di *filename* JavaScript. La voce deve essere una stringa *filename* o una tupla contenente la stringa *filename* e il dizionario *attributes*. Il *filename* deve essere relativo

a `html_static_path`, o un URI completo con uno schema come `https://example.org/script.js`. `attributes` viene usato per gli attributi del tag `<script>`. Il default è una lista vuota.

Esempio:

```
html_js_files = ['script.js',
                 'https://example.com/scripts/custom.js',
                 ('custom.js', {'async': 'async'})]
```

Come attributo speciale, `priority` può essere impostato come numero intero per caricare il file JavaScript in un passaggio precedente o più lento. Per ulteriori informazioni, fare riferimento a `Sphinx.add_js_file()`.

Added in version 1.8.

Cambiato nella versione 3.5: Supporto per l'attributo di priorità

`html_static_path`

Un elenco di path che contengono file statici personalizzati (come fogli di stile o file di script). I path relativi vengono considerati relativi alla directory di configurazione. Vengono copiati nella directory `_static` dell'output dopo i file statici del tema, quindi un file denominato `default.css` sovrascriverà il `default.css` del tema.

Poiché questi file non sono destinati ad essere creati, vengono automaticamente esclusi dai file sorgente.

Nota: Per ragioni di sicurezza, i dotfile sotto `html_static_path` non verranno copiati. Per copiarli intenzionalmente, si aggiunge ciascun filepath a questa impostazione:

```
html_static_path = ['_static', '_static/.htaccess']
```

Un altro modo per farlo, è quello di usare `html_extra_path`. Permette di copiare i dotfile nelle directory.

Cambiato nella versione 0.4: I percorsi in `html_static_path` ora possono contenere sottodirectory.

Cambiato nella versione 1.0: Le voci in `html_static_path` ora possono essere singoli file.

Cambiato nella versione 1.8: I file sotto `html_static_path` vengono esclusi dai sorgenti.

`html_extra_path`

Un elenco di path contenente file aggiuntivi non direttamente correlati alla documentazione, come `robots.txt` o `.htaccess`. I path relativi vengono considerati relativi alla directory di configurazione. Vengono copiati nella directory di output. Sovrascriveranno qualsiasi file esistente con lo stesso nome.

Poiché questi file non sono destinati ad essere creati, vengono automaticamente esclusi dai file sorgente.

Added in version 1.2.

Cambiato nella versione 1.4: I dotfile nella directory extra verranno copiati nella directory di output. E fa riferimento a *exclude_patterns* sulla copia di file e directory aggiuntivi e ignora se il percorso corrisponde ai pattern.

html_last_updated_fmt

Se questo non è None, viene inserito un timestamp 'Last updated on:' in fondo a ogni pagina, utilizzando il formato `strftime()`¹⁹⁵. La stringa vuota è equivalente a '%b %d, %Y' (o un equivalente di locale-dependent).

html_use_smartypants

Se true, virgolette e trattini vengono convertiti in entità tipograficamente corrette. Il default è: True.

Deprecato dalla versione 1.6: Per disabilitare le virgolette smart, utilizzare piuttosto *smartquotes*.

html_permalinks

Aggiunge ancoraggi di link per ogni ambiente di intestazione e descrizione. Il default è: True.

Added in version 3.5.

html_permalinks_icon

Testo per gli ancoraggi dei link per ciascun ambiente di intestazione e descrizione. Sono consentite entità HTML e Unicode. Il default è un segno di paragrafo; ¶

Added in version 3.5.

html_sidebars

I template della barra laterale custom, deve essere un dizionario che associa i nomi dei documenti ai nomi dei template.

Le chiavi possono contenere pattern in stile glob^{Pag. 94, 1}, nel qual caso tutti i documenti corrispondenti avranno le barre laterali specificate. (Viene emesso un warning quando più di un pattern di stile glob corrisponde a qualsiasi documento).

I valori possono essere liste o stringhe singole.

- Se un valore è una lista, specifica l'elenco completo dei template della barra laterale da includere. Se tutte o alcune delle barre laterali di default devono essere incluse, anch'esse devono essere inserite in questa lista.

Le barre laterali di default (per i documenti che non corrispondono ad alcun pattern) sono definite dal tema stesso. I temi nativi utilizzano questi template per default: ['localtoc.html', 'relations.html', 'sourcelink.html', 'searchbox.html'].

- Se un valore è una singola stringa, specifica una barra laterale custom da aggiungere tra le voci 'sourcelink.html' e 'searchbox.html'. Questo è per la compatibilità con le versioni di Sphinx precedenti alla 1.0.

¹⁹⁵ <https://docs.python.org/3/library/time.html#time.strftime>

Deprecato dalla versione 1.7: un singolo valore di stringa per `html_sidebars` verrà rimosso nella versione 2.0

I template di barra laterale nativi di cui è possibile eseguire il rendering sono:

- **localtoc.html** -- un sommario dettagliato del documento corrente
- **globaltoc.html** -- un sommario meno dettagliato per l'intero set di documentazione, compresso [collapsed]
- **relations.html** -- due link al documento precedente e al successivo
- **sourcelink.html** -- un link al sorgente del documento corrente, se abilitato in [html_show_sourcelink](#)
- **searchbox.html** -- la casella di "ricerca rapida"

Esempio:

```
html_sidebars = {
    '**': ['globaltoc.html', 'sourcelink.html', 'searchbox.html'],
    'using/windows': ['windowssidebar.html', 'searchbox.html'],
}
```

Questo visualizzerà il template custom `windowssidebar.html` e la casella di ricerca rapida all'interno della barra laterale del documento specificato e visualizzerà le barre laterali di default per tutte le altre pagine (tranne che il sommario locale viene sostituito da quello globale).

Added in version 1.0: La possibilità di utilizzare chiavi globbing e di specificare più barre laterali.

Si noti che questo valore non ha effetto solo se il tema scelto non possiede una barra laterale, come i temi nativi **scrolls** e **haiku**.

html_additional_pages

Ulteriori template di cui deve essere eseguito il rendering nelle pagine HTML devono essere un dizionario che associa i nomi dei documenti ai nomi dei template.

Esempio:

```
html_additional_pages = {
    'download': 'customdownload.html',
}
```

In questo modo il template `customdownload.html` verrà visualizzato come la pagina `download.html`.

html_domain_indices

Se `true`, genera indici specifici del dominio oltre a quello generale. Per es. il dominio Python, questo è l'indice globale del modulo. Il default è `True`.

Questo valore può essere un `bool` o un elenco di nomi di indice che dovrebbero essere generati. Per scoprire il nome dell'indice per un indice specifico, si guarda il nome del file HTML. Ad esempio, l'indice del modulo Python ha il nome `'py-modindex'`.

Added in version 1.0.

html_use_index

Se true, aggiunge un indice ai documenti HTML. Il default è True.

Added in version 0.4.

html_split_index

Se true, l'indice viene generato due volte: una volta come una singola pagina con tutte le voci e una volta come una pagina per lettera iniziale. Il default è False.

Added in version 0.4.

html_copy_source

Se true, i sorgenti reST vengono inclusi nella build HTML come `_sources/name`. Il default è True.

html_show_sourcelink

Se true (ed è true anche [html_copy_source](#)), i link ai sorgenti reST verranno aggiunti alla barra laterale. Il default è True.

Added in version 0.6.

html_sourcelink_suffix

Suffisso da aggiungere ai link dei sorgenti (vedere [html_show_sourcelink](#)), a meno che non abbiano già questo suffisso. Il default è `'.txt'`.

Added in version 1.5.

html_use_opensearch

Se non è vuoto, verrà generato un file di descrizione [OpenSearch](#)¹⁹⁶ e tutte le pagine conterranno un tag `<link>` che fa riferimento ad esso. Poiché OpenSearch non supporta URL relativi per la posizione della pagina di ricerca, il valore di questa opzione deve essere l'URL di base da cui vengono serviti questi documenti (senza barra finale), ad es. `"https://docs.python.org"`. Il default è `''`.

html_file_suffix

Questo è il suffisso per i file HTML generati, se impostato su un valore `str`¹⁹⁷. Se lasciato al valore None di default, il suffisso sarà `".html"`.

Added in version 0.4.

html_link_suffix

Suffisso per i link generati verso file HTML. Il default è quello su cui è impostato [html_file_suffix](#); può essere impostato in modo diverso (ad esempio per supportare diverse configurazioni del server web).

Added in version 0.6.

html_show_copyright

Se true, "(C) Copyright ..." viene visualizzato nel piè di pagina HTML. Il default è True.

¹⁹⁶ <https://github.com/dewitt/opensearch>

¹⁹⁷ <https://docs.python.org/3/library/stdtypes.html#str>

Added in version 1.0.

html_show_search_summary

Se true, il testo attorno alla parola chiave viene mostrato come riepilogo di ciascun risultato di ricerca. Il default è True.

Added in version 4.5.

html_show_sphinx

Se true, "Created using Sphinx" viene mostrato nel piè di pagina HTML. Il default è True.

Added in version 0.4.

html_output_encoding

Codifica dei file di output HTML. Il default è 'utf-8'. Notare che questa codifica deve essere sia un nome di codifica Python valido sia un valore charset HTML valido.

Added in version 1.0.

html_compact_lists

Se true, un elenco i cui elementi consistono in un singolo paragrafo e/o un sottoelenco i cui elementi ecc... (definizione ricorsiva) non utilizzerà l'elemento <p> per nessuno dei suoi elementi. Questo è il comportamento standard dei documenti. Il default è: True.

Added in version 1.0.

html_secnumber_suffix

Suffisso per la numerazione delle sezioni. Il default è: ". ". Impostare su " " per eliminare il punto finale sui numeri di sezione.

Added in version 1.0.

html_search_language

La lingua da utilizzare per generare l'indice di ricerca full-text HTML. Per default viene utilizzata la lingua globale selezionata con *language*. Se non è presente supporto per questa lingua, viene utilizzato "en" che seleziona la lingua inglese.

Il supporto è presente per queste lingue:

- da -- Danish
- nl -- Dutch
- en -- English
- fi -- Finnish
- fr -- French
- de -- German
- hu -- Hungarian
- it -- Italian
- ja -- Japanese
- no -- Norwegian

- `pt` -- Portuguese
- `ro` -- Romanian
- `ru` -- Russian
- `es` -- Spanish
- `sv` -- Swedish
- `tr` -- Turkish
- `zh` -- Chinese

Accelerare la velocità della build

Ogni lingua (tranne il giapponese) fornisce il proprio algoritmo di stemming. Sphinx utilizza un'implementazione Python per default. È possibile utilizzare un'implementazione C per accelerare la creazione del file dell'indice.

- [PorterStemmer](#)¹⁹⁸ (en)
- [PyStemmer](#)¹⁹⁹ (all languages)

Added in version 1.1: Con supporto per en e ja.

Cambiato nella versione 1.3: Aggiunte altre lingue.

`html_search_options`

Un dizionario con opzioni per il supporto della lingua di ricerca, vuoto per default. Il significato di queste opzioni dipende dalla lingua selezionata.

Il supporto inglese non ha opzioni.

Il supporto giapponese ha queste opzioni:

Type

`type` è una stringa del path del modulo punteggiata per specificare l'implementazione dello Splitter che dovrebbe essere derivato da `sphinx.search.ja.BaseSplitter`. Se non specificato o è `None`, verrà usato `'sphinx.search.ja.DefaultSplitter'`.

Si può scegliere tra questi moduli:

`'sphinx.search.ja.DefaultSplitter'`

Algoritmo TinySegmenter. Questo è lo splitter di default.

`'sphinx.search.ja.MecabSplitter'`

MeCab binding. Per utilizzare questo splitter, è necessario il binding python `'mecab'` o un link a una libreria dinamica (`'libmecab.so'` per linux, `'libmecab.dll'` per Windows).

¹⁹⁸ <https://pypi.org/project/PorterStemmer/>

¹⁹⁹ <https://pypi.org/project/PyStemmer/>

'sphinx.search.ja.JanomeSplitter'

Janome binding. Per usare questo splitter, è richiesto [Janome](https://pypi.org/project/Janome/)²⁰⁰.

Deprecato dalla versione 1.6: 'mecab', 'janome' e 'default' sono deprecati. Per mantenere la compatibilità, sono accettabili anche 'mecab', 'janome' e 'default'.

Altri valori di opzione dipendono dal valore dello splitter scelto.

Opzioni per 'mecab':

dic_enc

dic_enc option è la codifica per l'algoritmo MeCab.

dict

dict option è il dizionario da utilizzare per l'algoritmo MeCab.

lib

lib option il nome della libreria per trovare la libreria MeCab tramite ctypes se l'associazione Python non è installata.

Per esempio:

```
html_search_options = {
    'type': 'mecab',
    'dic_enc': 'utf-8',
    'dict': '/path/to/mecab.dic',
    'lib': '/path/to/libmecab.so',
}
```

Opzioni per 'janome':

user_dic

user_dic option è il percorso del file del dizionario utente per Janome.

user_dic_enc

user_dic_enc option è la codifica per il file del dizionario utente specificato dall'opzione user_dic. Il default è 'utf8'.

Added in version 1.1.

Cambiato nella versione 1.4: html_search_options per il giapponese è stato riorganizzato e qualsiasi splitter custom può essere utilizzato dalle impostazioni *type*.

Il supporto cinese ha queste opzioni:

- dict -- il percorso del dizionario jieba per usare il dizionario personalizzato.

html_search_scorer

Il nome di un file JavaScript (relativo alla directory di configurazione) che implementa un punteggio [scorer] dei risultati della ricerca. Se vuoto, verrà utilizzato il valore di default.

²⁰⁰ <https://pypi.org/project/Janome/>

Il valutatore (scorer) deve implementare la seguente interfaccia e, facoltativamente, può definire la funzione `score()` per un controllo più granulare.

```
const Scorer = {
  // Implement the following function to further tweak the score.
  →for each result
    score: result => {
      const [docName, title, anchor, descr, score, filename] =
      →result

      // ... calculate a new score ...
      return score
    },

    // query matches the full name of an object
    objNameMatch: 11,
    // or matches in the last dotted part of the object name
    objPartialMatch: 6,
    // Additive scores depending on the priority of the object
    objPrio: {
      0: 15, // used to be importantResults
      1: 5,  // used to be objectResults
      2: -5, // used to be unimportantResults
    },
    // Used when the priority is not in the mapping.
    objPrioDefault: 0,

    // query found in title
    title: 15,
    partialTitle: 7,

    // query found in terms
    term: 5,
    partialTerm: 2,
};
```

Added in version 1.2.

html_scaled_image_link

Se true, l'immagine stessa si linka a quella originale se non ha l'opzione 'target' le opzioni relative al dimensionamento: 'scale', 'width', 'height'. Il default è True.

Gli autori dei documenti possono disabilitare manualmente questa funzionalità fornendo la classe no-scaled-link all'immagine:

```
.. image:: sphinx.png
   :scale: 50%
   :class: no-scaled-link
```

Added in version 1.3.

Cambiato nella versione 3.0: È disabilitato per le immagini con classe `no-scaled-link`

html_math_renderer

Il nome dell'estensione `math_renderer` per l'output HTML. Il default è `'mathjax'`.

Added in version 1.8.

html_experimental_html5_writer

L'output viene elaborato con il writer HTML5. Il default è `False`.

Added in version 1.6.

Deprecato dalla versione 2.0.

html4_writer

L'output viene elaborato con il writer HTML4. Il default è `False`.

Opzioni per l'output HTML Singolo

singlehtml_sidebars

I template della barra laterale custom, deve essere un dizionario che associa i nomi dei documenti ai nomi dei template. E consente solo una chiave denominata `'index'`. Tutte le altre chiavi vengono ignorate. Per ulteriori informazioni, fare riferimento a [*html_sidebars*](#). Per default, è uguale a [*html_sidebars*](#).

Opzioni per l'output dell'help HTML

htmlhelp_basename

Nome base del file di output per il generatore dell'help HTML. Il default è `'pydoc'`.

htmlhelp_file_suffix

Questo è il suffisso del nome file per i file dell'help HTML generati. Il default è `".html"`.

Added in version 2.0.

htmlhelp_link_suffix

Suffisso per i link generati verso file HTML. Il default è `".html"`.

Added in version 2.0.

Opzioni per l'output dell'Help di Apple

Added in version 1.3.

Queste opzioni influenzano l'output dell'Help di Apple. Questo builder deriva da quello HTML, quindi anche le opzioni HTML si applicano dove appropriato.

Nota: L'output dell'Apple Help funzionerà solo su Mac OS X 10.6 e versioni successive, poiché richiede i tool a riga di comando **hiutil** e **codesign**, nessuno dei quali è Open Source.

Si può disabilitare l'uso di questi tool con [`applehelp_disable_external_tools`](#), ma il risultato non sarà un "help book" valido finché l'indicizzatore non verrà eseguito sulle cartelle `.lproj` nel bundle.

applehelp_bundle_name

Il nome base dell'Help Book di Apple. Il default è il nome [`project`](#).

applehelp_bundle_id

L'ID del bundle per l'help book bundle.

Avvertimento: Si *deve* impostare questo valore per generare l'Apple Help.

applehelp_dev_region

La regione di sviluppo. Il default è 'en-us', che è l'impostazione consigliata da Apple.

applehelp_bundle_version

La versione del bundle (come stringa). Il default è '1'.

applehelp_icon

L'icona dell'help bundle, o None per nessuna icona. Secondo la documentazione di Apple, dovrebbe trattarsi di una versione di 16 x 16 pixel dell'icona dell'applicazione con uno sfondo trasparente, salvata come file PNG.

applehelp_kb_product

Il tag del prodotto da utilizzare con [`applehelp_kb_url`](#). Il default è '`<project>-<release>`'.

applehelp_kb_url

L'URL del server della knowledge base, ad es. `https://example.com/kbsearch.py?p='product'&q='query'&l='lang'`. Help Viewer sostituirà i valori 'product', 'query' e 'lang' in fase di runtime con il contenuto di [`applehelp_kb_product`](#), rispettivamente il testo inserito dall'utente nella casella di ricerca e la lingua di sistema dell'utente.

Il default è None per nessuna ricerca remota.

applehelp_remote_url

L'URL per il contenuto remoto. Si può posizionare una copia della cartella Resources

dell'Help Book in questa posizione e l'Help Viewer tenterà di utilizzarla per recuperare il contenuto aggiornato.

per esempio. se si imposta su `https://example.com/help/Foo/` e l'Help Viewer vuole una copia di `index.html` per un utente inglese, cercherà in `https://example.com/help/Foo/en.lproj/index.html`.

Il default è `None` per nessun contenuto remoto.

applehelp_index_anchors

Se `True`, indica all'indicizzatore dell'help di indicizzare gli ancoraggi nell'HTML generato. Può risultare utile per passare a un argomento particolare utilizzando la funzione `AHLookupAnchor` o il metodo `openHelpAnchor.inBook:` nel codice. Consente anche di utilizzare URL `help:anchor;` consultare la documentazione Apple per ulteriori informazioni su questo argomento.

applehelp_min_term_length

Controlla la lunghezza minima del termine per l'indicizzatore dell'help. Il default è `None`, il che significa che verrà utilizzato il valore di default.

applehelp_stopwords

Una specifica della lingua (per utilizzare le stopwords native) o il path di una plist di stopwords oppure `None` se non si vogliono usare le stopwords. La plist di stopwords di default si trova su `/usr/share/hiutil/Stopwords.plist` e contiene, nel momento in cui si scrive, le stopwords per le seguenti lingue:

Lingua	Codice
English	en
German	de
Spanish	es
French	fr
Swedish	sv
Hungarian	hu
Italian	it

Il default è *language*, o, se non è impostato, `'en'`.

applehelp_locale

Specifica il [locale] per cui generare l'help. Viene utilizzato per determinare il nome della cartella `.lproj` nelle Resources dell'Help Book e passato all'indexer dell'help.

Il default è *language*, o, se non è impostato, `'en'`.

applehelp_title

Specifica il titolo dell'help book. Il default è `'<project> Help'`.

applehelp_codesign_identity

Specifica l'identità da utilizzare per la firma del codice oppure `None` se la firma del codice non deve essere eseguita.

Il default è il valore della variabile di ambiente `CODE_SIGN_IDENTITY`, impostata da Xcode per le fasi di creazione dello script, o `None` se tale variabile non è impostata.

applehelp_codesign_flags

Una *lista* di argomenti aggiuntivi da passare a **codesign** quando si firma l'help book.

Il default è un elenco basato sul valore della variabile di ambiente `OTHER_CODE_SIGN_FLAGS`, impostata da Xcode per le fasi di creazione dello script, o la lista vuota se tale variabile non è impostata.

applehelp_indexer_path

Il path del programma **hiutil**. Il default è `"/usr/bin/hiutil"`.

applehelp_codesign_path

Il path del programma **codesign**. Il default è `"/usr/bin/codesign"`.

applehelp_disable_external_tools

Se `True`, il builder non eseguirà l'indexer o il tool per la firma del codice, indipendentemente dalle altre impostazioni specificate.

Ciò è utile principalmente per i test o quando si esegue la build Sphinx su una piattaforma diversa da Mac OS X e quindi per completare i passaggi finali su OS X per qualche motivo.

Il default è `False`.

Opzioni per l'output epub

Queste opzioni influenzano l'output di epub. Poiché questo builder deriva da quello di HTML, anche le opzioni HTML si applicano dove appropriato. I valori effettivi per alcune opzioni non sono molto importanti, devono solo essere inseriti nei [Dublin Core metadata](#)²⁰¹.

epub_basename

Il nome base del file epub. Il default è il nome *project*.

epub_theme

Il tema HTML per l'output epub. Poiché i temi di default non sono ottimizzati per uno spazio ridotto sullo schermo, usare lo stesso tema per l'output HTML e l'epub, non è solitamente saggio. Il default, `'epub'`, è un tema progettato per risparmiare spazio visivo.

epub_theme_options

Un dictionary di opzioni che influenzano l'aspetto del tema selezionato. Questi sono specifici di ciascun tema. Per le opzioni dei temi nativi, consultare [questa sezione](#).

Added in version 1.2.

epub_title

Il titolo del documento. Per default è l'opzione *html_title* ma può essere impostata indipendentemente per la creazione di epub. Il default è l'opzione *project*.

Cambiato nella versione 2.0: Il default è l'opzione *project*.

²⁰¹ <https://dublincore.org/>

epub_description

La descrizione del documento. Il valore di default è 'unknown'.

Added in version 1.4.

Cambiato nella versione 1.5: Rinominato da `epub3_description`

epub_author

L'autore del documento. Viene inserito nei metadati Dublin Core. Il default è l'opzione `author`.

epub_contributor

Il nome di una persona, organizzazione, ecc. che ha svolto un ruolo secondario nella creazione del contenuto di una pubblicazione EPUB. Il valore di default è 'unknown'.

Added in version 1.4.

Cambiato nella versione 1.5: Rinominato da `epub3_contributor`

epub_language

La lingua del documento. Viene inserito nei metadati Dublin Core. Il default è l'opzione `language` o 'en' se non impostata.

epub_publisher

L'editore del documento. Viene inserito nei metadati Dublin Core. Puoi usare qualsiasi stringa sensata, ad es. la home page del progetto. Il default è l'opzione `author`.

epub_copyright

Il diritto d'autore del documento. Il default è l'opzione `copyright` ma può essere impostata indipendentemente per la creazione di epub.

epub_identifier

Un identificatore per il documento. Viene inserito nei metadati Dublin Core. Per i documenti pubblicati questo è il numero ISBN, ma si può anche utilizzare uno schema alternativo, ad es. la home page del progetto. Il valore di default è 'unknown'.

epub_scheme

Lo schema di pubblicazione per `epub_identifier`. Viene inserito nei metadati Dublin Core. Per i libri pubblicati lo schema è 'ISBN'. Se si usa la homepage del progetto, 'URL' sembra ragionevole. Il valore di default è 'unknown'.

epub_uid

Un identificatore univoco per il documento. Viene inserito nei metadati Dublin Core. È possibile utilizzare una stringa `XML's Name format`²⁰². Non è possibile utilizzare trattini, punti o numeri come primo carattere. Il valore di default è 'unknown'.

epub_cover

Le informazioni sulla copertina. Questa è una tupla contenente i nomi dei file dell'immagine di copertina e del template html. La copertina html renderizzata viene inserita come primo elemento nel dorso in `content.opf`. Se il nome del template è vuoto, non verrà creata alcuna copertina html. Se la tupla è vuota non viene creata alcuna copertina. Esempi:

²⁰² <https://www.w3.org/TR/REC-xml/#NT-NameStartChar>

```
epub_cover = ('_static/cover.png', 'epub-cover.html')
epub_cover = ('_static/cover.png', '')
epub_cover = ()
```

Il default è ().

Added in version 1.1.

epub_css_files

Un elenco di file CSS. La voce deve essere una stringa *filename* o una tupla contenente la stringa *filename* e il dizionario *attributes*. Per ulteriori informazioni, vedere [html_css_files](#).

Added in version 1.8.

epub_guide

Metadati per l'elemento guida di `content.opf`. Questa è una sequenza di tuple contenenti il *tipo*, l'*uri* e il *titolo* delle informazioni facoltative della guida. Consultare la documentazione OPF su <https://idpf.org/epub> per i dettagli. Se possibile, le voci di default per i tipi *cover* e *toc* vengono inserite automaticamente. Tuttavia, i tipi possono essere sovrascritti esplicitamente se le voci di default non sono appropriate. Esempio:

```
epub_guide = (('cover', 'cover.html', 'Cover Page'),)
```

Il default è ().

epub_pre_files

File aggiuntivi da inserire prima del testo generato da Sphinx. È un elenco di tuple contenente il nome del file e il titolo. Se il titolo è vuoto, non viene aggiunta alcuna voce a `toc.ncx`. Esempio:

```
epub_pre_files = [
    ('index.html', 'Welcome'),
]
```

Il default è [].

epub_post_files

File aggiuntivi da inserire dopo il testo generato da Sphinx. È un elenco di tuple contenente il nome del file e il titolo. Questa opzione può essere utilizzata per aggiungere un'appendice. Se il titolo è vuoto, non viene aggiunta alcuna voce a `toc.ncx`. Il default è [].

epub_exclude_files

Un elenco di file generati/copiati nella directory build ma che non devono essere inclusi nel file epub. Il default è [].

epub_tocdepth

La profondità del sommario nel file `toc.ncx`. Dovrebbe essere un numero intero maggiore di zero. Il default è 3. Nota: Un sommario molto annidato potrebbe essere difficile da consultare.

epub_tocdup

Questo flag determina se una voce del sommario viene re-inserita all'inizio del suo elenco nidificato. Consente una navigazione più semplice all'inizio di un capitolo, ma può creare confusione perché mescola voci di diversa profondità in un unico elenco. Il default è il valore `True`.

epub_tocscope

Questa impostazione controlla lo scope del sommario epub. L'impostazione può assumere i seguenti valori:

- `'default'` -- include tutte le voci del sommario che non sono nascoste (il default)
- `'includehidden'` -- include tutte le voci del sommario

Added in version 1.2.

epub_fix_images

Questo flag determina se Sphinx deve provare a correggere i formati delle immagini che non sono supportate da alcuni lettori di epub. Al momento vengono aggiornate le palette delle immagini con una piccola tabella dei colori. Per utilizzare questa opzione è necessario che Pillow, la Python Image Library, sia installata. Il default è `False` poiché la conversione automatica potrebbe perdere informazioni.

Added in version 1.2.

epub_max_image_width

Questa opzione specifica la larghezza massima delle immagini. Se è impostato su un valore maggiore di zero, le immagini con una larghezza maggiore del valore specificato verranno ridimensionate di conseguenza. Se è zero, non viene eseguito alcun ridimensionamento. Il default è `0`. Per utilizzare questa opzione è necessaria la Python Image Library (Pillow).

Added in version 1.2.

epub_show_urls

Controlla se visualizzare gli indirizzi URL. È molto utile per i lettori che non hanno altri mezzi per visualizzare l'URL linkata. Le impostazioni possono avere i seguenti valori:

- `'inline'` -- visualizza gli URL in linea tra parentesi (il default)
- `'footnote'` -- visualizza gli URL nelle note a piè di pagina
- `'no'` -- non visualizza gli URL

La visualizzazione degli URL in linea può essere personalizzata aggiungendo regole CSS per la classe `link-target`.

Added in version 1.2.

epub_use_index

Se `true`, si aggiunge un indice al documento epub. Il default è l'opzione `html_use_index` ma può essere impostata indipendentemente per la creazione di epub.

Added in version 1.2.

epub_writing_mode

Specifica la direzione della scrittura. Accetta 'horizontal' (il default) e 'vertical'

epub_writing_mode	'horizontal'	'vertical'
writing-mode ²⁰³	horizontal-tb	vertical-rl
page progression	left to right	right to left
iBook's Scroll Theme support	scroll-axis is vertical.	scroll-axis is horizontal.

Opzioni per l'output LaTeX

Queste opzioni riguardano l'output LaTeX.

latex_engine

Il motore LaTeX per creare i documenti. L'impostazione può assumere i seguenti valori:

- 'pdflatex' -- PDFLaTeX (il default)
- 'xelatex' -- XeLaTeX
- 'lualatex' -- LuaLaTeX
- 'platex' -- pLaTeX
- 'uplatex' -- upLaTeX (il default se *language* è 'ja')

Il supporto di 'pdflatex' per i caratteri Unicode è limitato.

Nota: 2.0 aggiunge al 'pdflatex' il supporto nei documenti in lingue Latine di lettere o parole occasionali in cirillico o greco. Questo non è automatico, vedere la discussione sulla chiave *latex_elements* 'fontenc'.

Se il progetto utilizza caratteri Unicode, impostare il motore su 'xelatex' o su 'lualatex' utilizzare un carattere OpenType con una copertura di glifi sufficientemente ampia si rivela spesso più semplice del cercare di far funzionare 'pdflatex' con i caratteri Unicode extra. A partire da Sphinx 2.0 il valore di default è GNU FreeFont che copre bene il latino, il cirillico e il greco.

Cambiato nella versione 2.1.0: Usa **xelatex** (e il pacchetto LaTeX xeCJK) per default per documenti in cinese.

Cambiato nella versione 2.2.1: Per default usa **xelatex** per i documenti in greco.

Cambiato nella versione 2.3: Aggiunge il supporto di **uplatex**.

Cambiato nella versione 4.0: **uplatex** diventa l'impostazione di default dei documenti giapponesi.

²⁰³ <https://developer.mozilla.org/en-US/docs/Web/CSS/writing-mode>

Contrariamente al rendering delle formule di *MathJaX math rendering in HTML output*, LaTeX richiede alcune configurazioni aggiuntive per supportare i letterali Unicode in *math*: l'unica soluzione completa (per quanto ne sappiamo) è usare 'xelatex' o 'lualatex' e aggiungere `r'\usepackage{unicode-math}`' (ad es. tramite la chiave *latex_elements* 'preamble'). Si potrebbe preferire `r'\usepackage[math-style=literal]{unicode-math}`' per mantenere un letterale Unicode come α (U+03B1) per esempio così com'è nell'output, anziché essere visualizzato come α .

latex_documents

Questo valore determina come raggruppare la struttura dei documenti nei sorgenti LaTeX. Deve essere una lista di tuple (*startdocname*, *targetname*, *title*, *author*, *theme*, *toctree_only*), dove gli elementi sono:

startdocname

Stringa che specifica il *document name* del documento master del file LaTeX. Tutti i documenti a cui fa riferimento *startdoc* negli alberi TOC verranno inclusi nel file LaTeX. (Per usare il documento root di default per la build LaTeX, si fornisce il proprio *root_doc*).

targetname

Nome del file LaTeX nella directory di output.

title

Titolo del documento LaTeX. Può essere vuoto per utilizzare il titolo del documento *startdoc*. Questo viene inserito come markup LaTeX, quindi i caratteri speciali come una barra rovesciata o una e commerciale devono essere rappresentati dai comandi LaTeX appropriati se devono essere inseriti letteralmente.

author

Autore del documento LaTeX. Si applica lo stesso avvertimento sul markup LaTeX di *title*. Si usa `\and` per separare più autori, come in: 'John `\and` Sarah' (le backslash devono essere sottoposte a escape Python per raggiungere LaTeX).

theme

Il tema LaTeX. Vedere *latex_theme*.

toctree_only

Dev'essere True o False. Se true, il documento *startdoc* stesso non è incluso nell'output, ma solo i documenti a cui fa riferimento tramite gli alberi TOC. Con questa opzione, si possono inserire elementi aggiuntivi nel documento master che vengono visualizzati nell'HTML, ma non nell'output LaTeX.

Added in version 1.2: In passato per includere la propria classe di documento era necessario anteporre al nome della classe di documento la stringa "sphinx". Questo non è più necessario.

Added in version 0.3: Il sesto elemento *toctree_only*. Sono ancora accettate tuple con 5 elementi.

latex_logo

Se fornito, deve essere il nome di un file immagine (relativo alla directory di configu-

razione) che rappresenta il logo dei documenti. È posizionato nella parte superiore del frontespizio. Il default è: None.

latex_toplevel_sectioning

Questo valore determina l'unità di sezionamento più in alta. Dovrebbe essere scelto tra 'part', 'chapter' o 'section'. Il default è None; l'unità di sezionamento più alta viene cambiata da documentclass: viene utilizzato section se documentclass è howto, altrimenti viene usato chapter.

Da notare che se LaTeX utilizza il comando `\part`, la numerazione delle unità di sezione a un livello di profondità non è sincronizzata con la numerazione HTML, poiché LaTeX numera continuamente `\chapter` (o `\section` per howto).

Added in version 1.4.

latex_appendices

Un elenco di nomi di documenti da aggiungere come appendice a tutti i manuali.

latex_domain_indices

Se true, genera indici specifici del dominio oltre a quello generale. Per es. il dominio Python, questo è l'indice globale del modulo. Il default è True.

Questo valore può essere un bool o un elenco di nomi di indici che dovrebbero essere generati, come per [html_domain_indices](#).

Added in version 1.0.

latex_show_pagerefs

Se true, aggiunge i riferimenti alla pagina dopo i riferimenti interni. Ciò è molto utile per le copie stampate del manuale. Il default è False.

Added in version 1.0.

latex_show_urls

Controlla se visualizzare gli indirizzi URL. Ciò è molto utile per le copie stampate del manuale. L'impostazione può assumere i seguenti valori:

- 'no' -- non visualizza gli URL (il default)
- 'footnote' -- visualizza gli URL nelle note a piè di pagina
- 'inline' -- visualizza gli URL in linea tra parentesi

Added in version 1.0.

Cambiato nella versione 1.1: Questo valore ora è una stringa; in precedenza era un valore booleano e un valore vero selezionava la visualizzazione 'inline'. Per compatibilità con le versioni precedenti, True è ancora accettato.

latex_use_latex_multicolumn

Il default è False: significa che le macro di Sphinx vengono utilizzate per le celle unite dalle tabelle della griglia. Consentono contenuti generali (blocchi letterali, elenchi, virgolette, ...) ma potrebbero avere problemi se la direttiva `tabularcolumns` è stata utilizzata per iniettare markup LaTeX del tipo `>{..}`, `<{..}`, `@{..}` come specifica della colonna.

Impostare su `True` significa utilizzare lo standard `\multicolumn` di LaTeX; questo è incompatibile con i blocchi letterali nella cella unita orizzontalmente e anche con più paragrafi in tale cella se la tabella viene visualizzata utilizzando `tabulary`.

Added in version 1.6.

`latex_table_style`

Una lista di classi di stile (stringhe). Attualmente supportate:

- `'booktabs'`: nessuna linea verticale e solo 2 o 3 linee orizzontali (quest'ultima se c'è un'intestazione), utilizzando il pacchetto `booktabs`²⁰⁴.
- `'borderless'`: nessuna linea.
- `'colorrows'`: le righe della tabella vengono visualizzate con colori di sfondo alternati. L'interfaccia per personalizzarle avviene tramite *chiavi dedicate* di *La configurazione di sphinxsetup*.

Importante: Con lo stile `'colorrows'`, il comando LaTeX `\rowcolors` diventa un "no-op" (questo comando ha delle limitazioni e non ha mai supportato correttamente tutti i tipi di tabelle che Sphinx produce in LaTeX). Aggiornare il progetto per utilizzare invece le chiavi *latex table color configuration*.

Il default è: `['booktabs', 'colorrows']`

Added in version 5.3.0.

Cambiato nella versione 6.0.0: Modifica del default da `[]` a `['booktabs', 'colorrows']`.

Ogni tabella può sovrascrivere lo stile globale tramite l'opzione `:class: option, o .. rst-class::` per tabelle senza direttive [no-directive] (cfr. *Tabelle*). Le classi attualmente riconosciute sono `booktabs`, `borderless`, `standard`, `colorrows`, `nocolorrows`. Le ultime due possono essere combinate con una qualsiasi delle prime tre. La classe `standard` produce tabelle con linee sia orizzontali che verticali (come è stato il default finora con Sphinx).

Una cella unita a più colonne a riga singola obbedirà al colore della riga, se impostato. Vedere anche `TableMergeColor{Header,Odd,Even}` nella sezione *La configurazione di sphinxsetup*.

Nota:

- In LaTeX è scolpito che una singola cella obbedirà al colore della riga anche se c'è un colore della colonna impostato tramite `\columncolor` da una specifica della colonna (vedere *tabularcolumns*). Sphinx fornisce `\sphinxnorowcolor` che può essere utilizzato in questo modo:

```
>{\columncolor{blue}\sphinxnorowcolor}
```

nella specifica di una colonna di tabella.

- Sphinx fornisce anche `\sphinxcolorblend` che tuttavia richiede Sphinx also provides `\sphinxcolorblend` che tuttavia richiede il pacchetto `xcolor`²⁰⁵. Ecco un esempio:

```
>\sphinxcolorblend{!95!red}}
```

Ciò significa che in questa colonna i colori delle righe saranno leggermente sfumati di rosso; fare riferimento alla documentazione di `xcolor`²⁰⁶ per maggiori informazioni sulla sintassi del suo comando `\blendcolors` (un `\blendcolors` al posto di `\sphinxcolorblend` modificherebbe i colori del *contenuto* della cella, non della cella *background colour panel...*). Si può trovare un esempio di utilizzo nella sezione *API deprecated* di questo documento in formato PDF.

Suggerimento: Per usare un colore speciale per il *contenuto* delle celle di una determinata colonna si usa `>\noindent\color{<color>}}`, possibilmente in aggiunta a quanto sopra.

- Le celle unite su più righe, sia a colonna singola che a più colonne, attualmente ignorano qualsiasi colonna, riga o colore di cella impostato.
 - È possibile per una cella semplice impostare un colore personalizzato tramite la direttiva `raw`²⁰⁷ e il comando LaTeX `\cellcolor` utilizzato ovunque nel contenuto della cella. Ciò attualmente non ha alcun effetto in una cella unita, qualunque sia il suo tipo.
-

Suggerimento: In un documento che non utilizza 'booktabs' globalmente, è possibile dare uno stile a una singola tabella tramite la classe `booktabs`, ma sarà necessario aggiungere `r'\usepackage{booktabs}` al preambolo LaTeX.

D'altra parte si può usare la classe `colorrows` per singole tabelle senza pacchetti aggiuntivi (poiché Sphinx dalla versione 5.3.0 carica sempre `colortbl`²⁰⁸).

latex_use_xindy

Se `True`, la creazione del PDF dai file LaTeX creati da Sphinx utilizzerà **xindy** (doc²⁰⁹) anziché **makeindex** per preparare l'indice dei termini generali (dall'uso di *index*). Ciò significa che le parole con caratteri UTF-8 verranno ordinate correttamente per *language*.

- Questa opzione viene ignorata se *latex_engine* è 'platex' (documenti giapponesi; **mendex** sostituisce poi **makeindex**).

²⁰⁴ <https://ctan.org/pkg/booktabs>

²⁰⁵ <https://ctan.org/pkg/xcolor>

²⁰⁶ <https://ctan.org/pkg/xcolor>

²⁰⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#raw>

²⁰⁸ <https://ctan.org/pkg/colortbl>

- Il default è True per 'xelatex' o 'lualatex' poiché **makeindex**, se qualsiasi termine indicizzato inizia con un carattere non ascii, crea file .ind contenenti byte non validi per la codifica UTF-8. Con 'lualatex' questo interrompe la creazione del PDF.
- Il default è False per 'pdflatex' ma True è consigliato per documenti non inglesi non appena alcuni termini indicizzati utilizzano caratteri non ASCII dallo script del linguaggio.

Sphinx aggiunge alla distribuzione base **xindy** un supporto dedicato per l'utilizzo del motore 'pdflatex' con script in cirillico. E sia con i motori 'pdflatex' che con i motori Unicode, i documenti in cirillico gestiscono correttamente l'indicizzazione dei nomi latini, anche con i segni diacritici.

Added in version 1.8.

latex_elements

Added in version 0.5.

La sua *documentazione* è stata spostata in *Personalizzazione di LaTeX*.

latex_docclass

Un dizionario che mappa 'howto' e 'manual' ai nomi di classi di documenti reali che verranno utilizzati come base per le due classi Sphinx. Il default consiste nell'usare 'article' per 'howto' e 'report' per 'manual'.

Added in version 1.0.

Cambiato nella versione 1.5: Nei documenti giapponesi (*language* è 'ja'), per default 'jreport' viene utilizzato per 'howto' e 'jsbook' per 'manual'.

latex_additional_files

Un elenco di nomi di file, relativi alla directory di configurazione, da copiare nella directory di build durante la creazione dell'output LaTeX. Ciò è utile per copiare file che Sphinx non copia automaticamente, ad es. se vengono referenziati nel LaTeX personalizzato aggiunto in `latex_elements`. I file immagine a cui si fa riferimento nei sorgenti (ad esempio tramite `.. image::`) vengono copiati automaticamente.

I nomi dei file non devono entrare in conflitto con quelli dei file copiati automaticamente.

Attenzione: I nomi dei file con estensione `.tex` verranno automaticamente passati al processo di creazione del PDF attivato da `sphinx-build -M latexpdf` o da `make latexpdf`. Se il file è stato aggiunto solo come `\input{}` da un preambolo modificato, è necessario aggiungere un ulteriore suffisso come `.txt` al nome del file e modificare di conseguenza il `\input{}` comando aggiunto al preambolo del documento LaTeX.

Added in version 0.6.

²⁰⁹ <https://xindy.sourceforge.net/>

Cambiato nella versione 1.2: Questo sovrascrive i file forniti da Sphinx come `sphinx.sty`.

latex_theme

Il "tema" che dovrebbe utilizzare l'output LaTeX. È una raccolta di impostazioni per l'output LaTeX (es. la classe del documento, l'unità di sezionamento di livello superiore e così via).

Come temi nativi di LaTeX, `manual` e `howto` sono [bundled].

manual

Un tema LaTeX per scrivere un manuale. Importa la classe di documenti `report` (i documenti giapponesi utilizzano `jsbook`).

howto

Un tema LaTeX per scrivere un articolo. Importa la classe del documento `article` (i documenti giapponesi usano invece `jreport`). [*latex_appendices*](#) è disponibile solo per questo tema.

Il default è `'manual'`.

Added in version 3.0.

latex_theme_options

Un dictionary di opzioni che influenzano l'aspetto del tema selezionato.

Added in version 3.1.

latex_theme_path

Un elenco di path che contengono temi LaTeX personalizzati come sottodirectory. I path relativi vengono considerati relativi alla directory di configurazione.

Added in version 3.0.

Opzioni per l'output text

Queste opzioni influenzano l'output text.

text_newlines

Determina quali caratteri di fine riga vengono utilizzati nell'output del testo.

- `'unix'`: utilizza terminazioni di riga in stile Unix (`\n`)
- `'windows'`: utilizza terminazioni di riga in stile Windows (`\r\n`)
- `'native'`: utilizza lo stile di fine riga della piattaforma su cui è costruita la documentazione

Il default è: `'unix'`.

Added in version 1.1.

text_sectionchars

Una stringa di 7 caratteri da utilizzare per delineare le sezioni. Il primo carattere viene utilizzato per i titoli di primo livello, il secondo per i titoli di secondo livello e così via.

Il default è `'*=-~"+` '`.

Added in version 1.1.

text_add_secnumbers

Un valore booleano che decide se i numeri di sezione sono inclusi nell'output del testo. Il default è `True`.

Added in version 1.7.

text_secnumber_suffix

Suffisso per i numeri di sezione nell'output del testo. Il default è: `". "`. Impostare su `"` per eliminare il punto finale sui numeri di sezione.

Added in version 1.7.

Opzioni per l'output della pagina del manuale

Queste opzioni influenzano l'output della pagina del manuale.

man_pages

Questo valore determina come raggruppare l'albero del documento in pagine di manuale. Deve essere una lista di tuple (`startdocname`, `name`, `description`, `authors`, `section`), dove gli elementi sono:

startdocname

Stringa che specifica il *document name* del documento master della pagina del manuale. Tutti i documenti a cui fa riferimento il documento *startdoc* negli alberi TOC verranno inclusi nel file del manuale. (Per utilizzare il documento root di default per la creazione delle pagine del manuale, utilizzare il proprio *root_doc*).

name

Nome della pagina del manuale. Dovrebbe essere una stringa breve senza spazi o caratteri speciali. Viene utilizzato per determinare il nome del file e il nome della pagina del manuale (nella sezione NAME).

description

Descrizione della pagina del manuale. Viene utilizzato nella sezione NAME. Può essere una stringa vuota, per non generare automaticamente la sezione NAME.

authors

Una lista di stringhe con autori o una singola stringa. Può essere una stringa o una lista vuota se non si vuole generare automaticamente una sezione AUTHORS nella pagina man.

section

La sezione della pagina del manuale. Utilizzato per il nome del file di output e nell'intestazione della pagina del manuale.

Added in version 1.0.

man_show_urls

Se true, aggiunge gli indirizzi URL dopo i collegamenti. Il default è False.

Added in version 1.1.

man_make_section_directory

Se true, crea una directory di sezione sulla build della pagina man. Il default è True.

Added in version 3.3.

Cambiato nella versione 4.0: Il default è cambiato da False a True.

Cambiato nella versione 4.0.2: Il default è cambiato in True da False di nuovo.

Opzioni per l'output Texinfo

Queste opzioni influenzano l'output di Texinfo.

texinfo_documents

Questo valore determina come raggruppare la struttura del documento nei file sorgenti di Texinfo. Deve essere una lista di tuple (`startdocname`, `targetname`, `title`, `author`, `dir_entry`, `description`, `category`, `toctree_only`), dove gli elementi sono:

startdocname

Stringa che specifica il *document name* del documento master del file Texinfo. Tutti i documenti a cui fa riferimento *startdoc* negli alberi TOC saranno inclusi nel file Texinfo. (Per utilizzare il documento master di default per la build Texinfo, si fornisce il proprio *root_doc*).

targetname

Nome del file (senza estensione) del file Texinfo nella directory di output.

title

Titolo del documento Texinfo. Può essere vuoto per utilizzare il titolo del documento *startdoc*. Inserito come markup Texinfo, quindi i caratteri speciali come @ e {} dovranno essere sottoposti a escape per essere inseriti letteralmente.

author

Autore del documento Texinfo. Inserito come markup Texinfo. Si usa @* per separare più autori, come in: 'John@*Sarah'.

dir_entry

Il nome che apparirà nel file di menù DIR di livello superiore.

description

Testo descrittivo da visualizzare nel file di menù DIR di livello superiore.

category

Specifica la sezione in cui questa voce apparirà nel file di menù DIR di livello superiore.

toctree_only

Dev'essere True o False. Se true, il documento *startdoc* stesso non è incluso nell'output, ma solo i documenti a cui fa riferimento tramite gli alberi TOC. Con questa opzione, si possono inserire elementi aggiuntivi nel documento master che vengono visualizzati nell'HTML, ma non nell'output di Texinfo.

Added in version 1.1.

texinfo_appendices

Un elenco di nomi di documenti da aggiungere come appendice a tutti i manuali.

Added in version 1.1.

texinfo_domain_indices

Se true, genera indici specifici del dominio oltre a quello generale. Per es. il dominio Python, questo è l'indice globale del modulo. Il default è True.

Questo valore può essere un bool o un elenco di nomi di indici che dovrebbero essere generati, come per [html_domain_indices](#).

Added in version 1.1.

texinfo_show_urls

Controlla come visualizzare gli indirizzi URL.

- 'footnote' -- visualizza gli URL nelle note a piè di pagina (il default)
- 'no' -- non visualizza gli URL
- 'inline' -- visualizza gli URL in linea tra parentesi

Added in version 1.1.

texinfo_no_detailmenu

Se true, non genera un @detailmenu nel menù del nodo "Top" contenente voci per ciascun sottonodo nel documento. Il default è False.

Added in version 1.2.

texinfo_elements

Un dizionario che contiene frammenti di Texinfo che sovrascrivono quelli di Sphinx viene solitamente inserito nei file `.texi` generati.

- Tra le chiavi sovrascrivibili ci sono:

'paragraphindent'

Numero di spazi per il rientro della prima riga di ogni paragrafo, il default è 2. Specificare 0 per nessuna indentazione.

'exampleindent'

Numero di spazi per indentare le righe per gli esempi o i blocchi letterali, il default è 4. Specificare 0 per nessuna indentazione.

'preamble'

Il markup Texinfo inserito nei pressi dell'inizio del file.

'copying'

Markup Texinfo inserito nel blocco @copying e visualizzato dopo il titolo. Il default è costituito da un semplice frontespizio che identifica il progetto.

- Le chiavi impostate da altre opzioni e che pertanto non devono essere sovrascritte sono:

'author' 'body' 'date' 'direntry' 'filename' 'project' 'release'
'title'

Added in version 1.1.

texinfo_cross_references

Se false, non vengono generati riferimenti in linea in un documento. Ciò rende un file info più leggibile con lo stand-alone (info). Il default è True.

Added in version 4.4.

Opzioni per l'output QtHelp

Queste opzioni riguardano l'output di qthelp. Poiché questo builder deriva da quello di HTML, anche le opzioni HTML si applicano dove appropriato.

qthelp_basename

Il nome base del file qthelp. Il default è il nome *project*.

qthelp_namespace

Lo spazio dei nomi per il file qthelp. Il default è org.sphinx.<project_name>.<project_version>.

qthelp_theme

Il tema HTML per l'output qthelp. Il default è 'nonav'.

qthelp_theme_options

Un dictionary di opzioni che influenzano l'aspetto del tema selezionato. Questi sono specifici di ciascun tema. Per le opzioni dei temi nativi, consultare *questa sezione*.

Opzioni per il builder di linkcheck

linkcheck_ignore

Un elenco di espressioni regolari che corrispondono agli URI che non devono essere controllati durante una build linkcheck. Esempio:

```
linkcheck_ignore = [r'https://localhost:\d+/']
```

Added in version 1.1.

linkcheck_allowed_redirects

Un dizionario che mappa un pattern dell'URI dorgente a un pattern dell'URI canonico. Il builder linkcheck considera il link reindirizzato come "funzionante" quando:

- il link nel documento corrisponde al pattern URI sorgente e
- la posizione di reindirizzamento corrisponde al pattern dell'URI canonico.

Esempio:

```
linkcheck_allowed_redirects = {  
    # All HTTP redirections from the source URI to the canonical_  
    ↪URI will be treated as "working".  
    r'https://sphinx-doc\.org/.*': r'https://sphinx-doc\.org/en/  
    ↪master/.*' }  
}
```

Se impostato, il builder linkcheck emetterà un warning quando viene trovato un reindirizzamento non consentito. È utile rilevare reindirizzamenti imprevisti con *the warn-is-error mode*.

Added in version 4.1.

linkcheck_request_headers

Un dizionario che associa gli URL di base agli header delle richieste HTTP.

La chiave è una stringa di base dell'URL come "https://www.sphinx-doc.org/". Per specificare header per altri host, si può utilizzare "*". Corrisponde a tutti gli host solo quando l'URL non corrisponde ad altre impostazioni.

Il valore è un dizionario che associa il nome dell'header al relativo valore.

Esempio:

```
linkcheck_request_headers = {  
    "https://www.sphinx-doc.org/": {  
        "Accept": "text/html",  
        "Accept-Encoding": "utf-8",  
    },  
    "*": {  
        "Accept": "text/html,application/xhtml+xml",  
    }  
}
```

Added in version 3.1.

linkcheck_retries

Il numero di volte in cui il builder linkcheck tenterà di controllare un URL prima di dichiararlo non funzionante. Il default è 1 tentativo.

Added in version 1.4.

linkcheck_timeout

La durata, in secondi, di attesa di una risposta da parte del builder linkcheck dopo ogni richiesta di link ipertestuale. Il default è 30 secondi.

Added in version 1.1.

linkcheck_workers

Il numero di thread worker da utilizzare durante il controllo dei link. Il default è 5 thread.

Added in version 1.1.

linkcheck_anchors

Se true, controlla la validità degli `#anchor` nei link. Poiché ciò richiede il download dell'intero documento, si rallenta notevolmente quando abilitato. Il default è True.

Added in version 1.2.

linkcheck_anchors_ignore

Una lista di espressioni regolari che corrispondono alle [anchor] Sphinx dovrebbe saltate controllandone la validità nei link. Ciò consente di saltare gli ancoraggi aggiunti dal JavaScript di un sito Web per controllare le pagine dinamiche o quando si attiva una richiesta REST interna. Il default è `["^!"]`.

Suggerimento: Usare `linkcheck_anchors_ignore_for_url` per controllare un URL, ma evita di verificare l'esistenza degli ancoraggi.

Nota: Per ignorare gli ancoraggi di una pagina specifica o di pagine che corrispondono a un pattern specifico (ma controlla comunque le occorrenze delle stesse pagine che non hanno ancora), usare invece `linkcheck_ignore`, ad esempio come segue:

```
linkcheck_ignore = [  
    'https://www.sphinx-doc.org/en/1.7/intro.html#',  
]
```

Added in version 1.5.

linkcheck_anchors_ignore_for_url

Una lista o una tupla di espressioni regolari che corrispondono agli URL per i quali Sphinx non deve verificare la validità degli ancoraggi. Ciò consente di saltare i controlli di ancoraggio per pagina pur controllando comunque la validità della pagina stessa. Il default è una tupla vuota `()`.

Added in version 7.1.

linkcheck_auth

Passa le informazioni di autenticazione eseguendo una build `linkcheck`.

Una lista di tuple (`regex_pattern`, `auth_info`) in cui gli elementi sono:

regex_pattern

Un'espressione regolare che corrisponde a un URI.

auth_info

Informazioni di autenticazione da utilizzare per quell'URI. Il valore può essere qualsiasi cosa compresa dalla libreria `requests` (consultare [requests Authentication](#)²¹⁰ per i dettagli).

Il builder `linkcheck` utilizzerà il primo valore `auth_info` corrispondente che riesce a trovare nella lista `linkcheck_auth`, quindi i valori precedenti nella lista hanno valori più alti di priorità.

Esempio:

```
linkcheck_auth = [  
    ('https://foo\yourcompany\com/.+', ('johndoe', 'secret')),  
    ('https://.+\.yourcompany\com/.+', HTTPDigestAuth(...)),  
]
```

Added in version 2.3.

linkcheck_rate_limit_timeout

Il builder `linkcheck` può inviare un gran numero di richieste allo stesso sito in un breve periodo di tempo. Questa impostazione regola il comportamento del builder quando i server indicano che le richieste hanno una velocità limitata.

Se un server indica quando riprovare (utilizzando l'intestazione `Retry-After`²¹¹), `linkcheck` segue sempre l'indicazione del server.

Altrimenti, `linkcheck` attende un minuto prima di riprovare e continua a raddoppiare il tempo di attesa tra i tentativi finché non riesce o supera il `linkcheck_rate_limit_timeout`. Per default, il timeout è di 300 secondi. I timeout personalizzati devono essere espressi in numeri di secondi.

Added in version 3.4.

linkcheck_exclude_documents

Una lista di espressioni regolari che corrispondono a documenti in cui Sphinx non deve verificare la validità dei link. Può essere utilizzato per consentire il decadimento dei link nelle sezioni legacy o storiche della documentazione.

Esempio:

```
# ignore all links in documents located in a subfolder named  
↳ 'legacy'  
linkcheck_exclude_documents = [r'*/legacy/.*']
```

Added in version 4.4.

linkcheck_allow_unauthorized

Quando un server web risponde con HTTP 401 (non autorizzata), l'attuale comportamento di default di Sphinx è quello di considerare il link come "working" (funzionante). Per modificare questo comportamento, impostare questa opzione su `False`.

Il default per questa opzione verrà modificato in Sphinx 8.0; da quella versione in poi, le risposte HTTP 401 ai link ipertestuali controllati verranno trattate come "broken" (interrotte) per default.

Added in version 7.3.

²¹⁰ <https://requests.readthedocs.io/en/latest/user/authentication/#authentication>

²¹¹ <https://datatracker.ietf.org/doc/html/rfc7231#section-7.1.3>

linkcheck_report_timeouts_as_broken

Quando una risposta HTTP non viene ricevuta da un server web prima della scadenza del `linkcheck_timeout` configurato, l'attuale comportamento di default di Sphinx è quello di trattare il link come 'broken' (interrotto). Per segnalare i timeout utilizzando un codice di report distinto di timeout, si imposta `linkcheck_report_timeouts_as_broken` a False.

Da Sphinx 8.0 in poi, i timeout che si verificano durante il controllo dei link ipertestuali verranno segnalati utilizzando il nuovo codice di stato.

Added in version 7.3.

Opzioni per il builder XML

xml_pretty

Se true, stampa 'pretty' l'XML. Il default è True.

Added in version 1.2.

Opzioni per il dominio C

c_id_attributes

Una lista di stringhe che il parser inoltre dovrebbe accettare come attributi. Questo può essere utilizzato, ad esempio, quando gli attributi sono stati definiti con `#define` per la portabilità.

Added in version 3.0.

c_paren_attributes

Una lista di stringhe che il parser inoltre dovrebbe accettare come attributi con un argomento. Cioè, se `my_align_as` è nella lista, allora `my_align_as(X)` viene analizzato come un attributo per tutte le stringhe X che hanno parentesi bilanciate (`()`, `[]`, and `{}`). Questo può essere utilizzato, ad esempio, quando gli attributi sono stati definiti con `#define` per la portabilità.

Added in version 3.0.

c_extra_keywords

Una lista di identificatori che devono essere riconosciuti come parole chiave dal parser C. Il default è `['alignas', 'alignof', 'bool', 'complex', 'imaginary', 'noreturn', 'static_assert', 'thread_local']`.

Added in version 4.0.3.

c_maximum_signature_line_length

Se la lunghezza in caratteri di una firma supera il numero impostato, ogni parametro verrà visualizzato su una singola riga logica. Questa è un'impostazione specifica del dominio, che sovrascrive `maximum_signature_line_length`.

Added in version 7.1.

Opzioni per il dominio C++

`cpp_index_common_prefix`

Una lista di prefissi che verranno ignorati durante l'ordinamento degli oggetti C++ nell'indice globale. Per esempio `['awesome_lib::']`.

Added in version 1.5.

`cpp_id_attributes`

Una lista di stringhe che il parser inoltre dovrebbe accettare come attributi. Questo può essere utilizzato, ad esempio, quando gli attributi sono stati definiti con `#define` per la portabilità.

Added in version 1.5.

`cpp_paren_attributes`

Una lista di stringhe che il parser inoltre dovrebbe accettare come attributi con un argomento. Cioè, se `my_align_as` è nella lista, allora `my_align_as(X)` viene analizzato come un attributo per tutte le stringhe `X` che hanno parentesi bilanciate (`(`), `[`), and `{}`). Questo può essere utilizzato, ad esempio, quando gli attributi sono stati definiti con `#define` per la portabilità.

Added in version 1.5.

`cpp_maximum_signature_line_length`

Se la lunghezza in caratteri di una firma supera il numero impostato, ogni parametro verrà visualizzato su una singola riga logica. Questa è un'impostazione specifica del dominio, che sovrascrive `maximum_signature_line_length`.

Added in version 7.1.

Opzioni per il dominio Python

`python_display_short_literal_types`

Questo valore controlla il modo in cui vengono visualizzati i tipi `Literal`²¹². L'impostazione è booleana, il default è `False`.

Esempi

Gli esempi seguenti utilizzano la seguente direttiva `py:function`:

```
.. py:function:: serve_food(item: Literal["egg", "spam", "lobster_
→thermidor"]) -> None
```

Quando è `False`, i tipi `Literal`²¹³ vengono visualizzati secondo la sintassi Python standard, ovvero:

```
serve_food(item: Literal["egg", "spam", "lobster thermidor
→"]) -> None
```

Quando è True, i tipi `Literal`²¹⁴ vengono visualizzati con una sintassi breve ispirata a `PEP 604`²¹⁵, ovvero:

```
serve_food(item: "egg" | "spam" | "lobster thermidor") ->
↳ None
```

Added in version 6.2.

`python_use_unqualified_type_names`

Se true, sopprime il nome del modulo del riferimento Python se può essere risolto. Il default è False.

Added in version 4.0.

Nota: Questa configurazione è ancora in fase sperimentale

`python_maximum_signature_line_length`

Se la lunghezza in caratteri di una firma supera il numero impostato, ogni argomento o parametro di tipo verrà visualizzato su una singola riga logica. Questa è un'impostazione specifica del dominio, che sovrascrive `maximum_signature_line_length`.

Per il dominio Python, la lunghezza della firma dipende dalla formattazione dei parametri di tipo o dell'elenco di argomenti. Nel primo caso, la lunghezza della firma ignora la lunghezza dell'elenco degli argomenti; per quest'ultimo, la lunghezza della firma ignora la lunghezza dell'elenco dei parametri di tipo.

Ad esempio, con `python_maximum_signature_line_length = 20`, solo l'elenco dei parametri di tipo verrà racchiuso mentre l'elenco degli argomenti verrà visualizzato su una singola riga

```
.. py:function:: add[T: VERY_LONG_SUPER_TYPE, U: VERY_LONG_SUPER_
↳ TYPE](a: T, b: U)
```

Added in version 7.1.

Opzioni per il dominio Javascript

`javascript_maximum_signature_line_length`

Se la lunghezza in caratteri di una firma supera il numero impostato, ogni parametro verrà visualizzato su una singola riga logica. Questa è un'impostazione specifica del dominio, che sovrascrive `maximum_signature_line_length`.

Added in version 7.1.

²¹² <https://docs.python.org/3/library/typing.html#typing.Literal>

²¹³ <https://docs.python.org/3/library/typing.html#typing.Literal>

²¹⁴ <https://docs.python.org/3/library/typing.html#typing.Literal>

²¹⁵ <https://peps.python.org/pep-0604/>

Esempio di file di configurazione

```
# test documentation build configuration file, created by
# sphinx-quickstart on Sun Jun 26 00:00:43 2016.
#
# This file is executed through importlib.import_module with
# the current directory set to its containing dir.
#
# Note that not all possible configuration values are present in this
# autogenerated file.
#
# All configuration values have a default; values that are commented
# out
# serve to show the default.

# If extensions (or modules to document with autodoc) are in another
# directory,
# add these directories to sys.path here. If the directory is relative
# to the
# documentation root, use os.path.abspath to make it absolute, like
# shown here.
#
# import os
# import sys
# sys.path.insert(0, os.path.abspath('.'))

# -- General configuration -----
# ----

# If your documentation needs a minimal Sphinx version, state it here.
#
# needs_sphinx = '1.0'

# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
extensions = []

# Add any paths that contain templates here, relative to this
# directory.
templates_path = ['_templates']

# The suffix(es) of source filenames.
# You can specify multiple suffix as a list of string:
#
# source_suffix = ['.rst', '.md']
source_suffix = '.rst'
```

(continues on next page)

(continua dalla pagina precedente)

```
# The encoding of source files.
#
# source_encoding = 'utf-8-sig'

# The master toctree document.
root_doc = 'index'

# General information about the project.
project = 'test'
copyright = '2016, test'
author = 'test'

# The version info for the project you're documenting, acts as
→ replacement for
# |version| and |release|, also used in various other places
→ throughout the
# built documents.
#
# The short X.Y version.
version = 'test'
# The full version, including alpha/beta/rc tags.
release = 'test'

# The language for content autogenerated by Sphinx. Refer to
→ documentation
# for a list of supported languages.
#
# This is also used if you do content translation via gettext catalogs.
# Usually you set "language" from the command line for these cases.
language = None

# There are two options for replacing |today|: either, you set today
→ to some
# non-false value, then it is used:
#
# today = "
#
# Else, today_fmt is used as the format for a strftime call.
#
# today_fmt = '%B %d, %Y'

# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# These patterns also affect html_static_path and html_extra_path
exclude_patterns = ['_build', 'Thumbs.db', '.DS_Store']
```

(continues on next page)

(continua dalla pagina precedente)

```

# The reST default role (used for this markup: `text`) to use for all
# documents.
#
# default_role = None

# If true, '()' will be appended to :func: etc. cross-reference text.
#
# add_function_parentheses = True

# If true, the current module name will be prepended to all description
# unit titles (such as .. function::).
#
# add_module_names = True

# If true, sectionauthor and moduleauthor directives will be shown in
→the
# output. They are ignored by default.
#
# show_authors = False

# The name of the Pygments (syntax highlighting) style to use.
pygments_style = 'sphinx'

# A list of ignored prefixes for module index sorting.
# modindex_common_prefix = []

# If true, keep warnings as "system message" paragraphs in the built
→documents.
# keep_warnings = False

# If true, `todo` and `todoList` produce output, else they produce
→nothing.
todo_include_todos = False

# -- Options for HTML output -----
→----

# The theme to use for HTML and HTML Help pages.  See the
→documentation for
# a list of builtin themes.
#
html_theme = 'alabaster'

# Theme options are theme-specific and customize the look and feel of

```

(continues on next page)

(continua dalla pagina precedente)

```
→a theme
# further. For a list of options available for each theme, see the
# documentation.
#
# html_theme_options = {}

# Add any paths that contain custom themes here, relative to this_
→directory.
# html_theme_path = []

# The name for this set of Sphinx documents.
# "<project> v<release> documentation" by default.
#
# html_title = 'test vtest'

# A shorter title for the navigation bar. Default is the same as html_
→title.
#
# html_short_title = None

# The name of an image file (relative to this directory) to place at_
→the top
# of the sidebar.
#
# html_logo = None

# The name of an image file (relative to this directory) to use as a_
→favicon of
# the docs. This file should be a Windows icon file (.ico) being_
→16x16 or 32x32
# pixels large.
#
# html_favicon = None

# Add any paths that contain custom static files (such as style_
→sheets) here,
# relative to this directory. They are copied after the builtin static_
→files,
# so a file named "default.css" will overwrite the builtin "default.css"
→".
html_static_path = ['_static']

# Add any extra paths that contain custom files (such as robots.txt or
# .htaccess) here, relative to this directory. These files are copied
# directly to the root of the documentation.
#
```

(continues on next page)

(continua dalla pagina precedente)

```
# html_extra_path = []

# If not None, a 'Last updated on:' timestamp is inserted at every page
# bottom, using the given strftime format.
# The empty string is equivalent to '%b %d, %Y'.
#
# html_last_updated_fmt = None

# Custom sidebar templates, maps document names to template names.
#
# html_sidebars = {}

# Additional templates that should be rendered to pages, maps page_
# names to
# template names.
#
# html_additional_pages = {}

# If false, no module index is generated.
#
# html_domain_indices = True

# If false, no index is generated.
#
# html_use_index = True

# If true, the index is split into individual pages for each letter.
#
# html_split_index = False

# If true, links to the reST sources are added to the pages.
#
# html_show_sourcelink = True

# If true, "Created using Sphinx" is shown in the HTML footer. Default_
# is True.
#
# html_show_sphinx = True

# If true, "(C) Copyright ..." is shown in the HTML footer. Default is_
# True.
#
# html_show_copyright = True

# If true, an OpenSearch description file will be output, and all_
# pages will
```

(continues on next page)

(continua dalla pagina precedente)

```

# contain a <link> tag referring to it. The value of this option must
→be the
# base URL from which the finished HTML is served.
#
# html_use_opensearch = "

# This is the file name suffix for HTML files (e.g. ".xhtml").
# html_file_suffix = None

# Language to be used for generating the HTML full-text search index.
# Sphinx supports the following languages:
#   'da', 'de', 'en', 'es', 'fi', 'fr', 'hu', 'it', 'ja'
#   'nl', 'no', 'pt', 'ro', 'ru', 'sv', 'tr', 'zh'
#
# html_search_language = 'en'

# A dictionary with options for the search language support, empty by
→default.
# 'ja' uses this config value.
# 'zh' user can custom change `jieba` dictionary path.
#
# html_search_options = {'type': 'default'}

# The name of a javascript file (relative to the configuration
→directory) that
# implements a search results scorer. If empty, the default will be
→used.
#
# html_search_scorer = 'scorer.js'

# Output file base name for HTML help builder.
htmlhelp_basename = 'testdoc'

# -- Options for LaTeX output -----
→-----

latex_elements = {
    # The paper size ('letterpaper' or 'a4paper').
    #
    # 'papersize': 'letterpaper',

    # The font size ('10pt', '11pt' or '12pt').
    #
    # 'pointsize': '10pt',

    # Additional stuff for the LaTeX preamble.

```

(continues on next page)

(continua dalla pagina precedente)

```

#
# 'preamble': "",

# Latex figure (float) alignment
#
# 'figure_align': 'htbp',
}

# Grouping the document tree into LaTeX files. List of tuples
# (source start file, target name, title,
#  author, documentclass [howto, manual, or own class]).
latex_documents = [
    (root_doc, 'test.tex', 'test Documentation',
     'test', 'manual'),
]

# The name of an image file (relative to this directory) to place at
→ the top of
# the title page.
#
# latex_logo = None

# If true, show page references after internal links.
#
# latex_show_pagerefs = False

# If true, show URL addresses after external links.
#
# latex_show_urls = False

# Documents to append as an appendix to all manuals.
#
# latex_appendices = []

# If false, no module index is generated.
#
# latex_domain_indices = True


# -- Options for manual page output -----
→ ----

# One entry per manual page. List of tuples
# (source start file, name, description, authors, manual section).
man_pages = [
    (root_doc, 'test', 'test Documentation',

```

(continues on next page)

(continua dalla pagina precedente)

```

    [author], 1)
]

# If true, show URL addresses after external links.
#
# man_show_urls = False

# -- Options for Texinfo output -----
→-----

# Grouping the document tree into Texinfo files. List of tuples
# (source start file, target name, title, author,
#  dir menu entry, description, category)
texinfo_documents = [
    (root_doc, 'test', 'test Documentation',
     author, 'test', 'One line description of project.',
     'Miscellaneous'),
]

# Documents to append as an appendix to all manuals.
#
# texinfo_appendices = []

# If false, no module index is generated.
#
# texinfo_domain_indices = True

# How to display URL addresses: 'footnote', 'no', or 'inline'.
#
# texinfo_show_urls = 'footnote'

# If true, do not generate a @detailmenu in the "Top" node's menu.
#
# texinfo_no_detailmenu = False

# If false, do not generate in manual @ref nodes.
#
# texinfo_cross_references = False

# -- A random example -----
→-----

import sys, os
sys.path.insert(0, os.path.abspath('.'))
exclude_patterns = ['zzz']

```

(continues on next page)

(continua dalla pagina precedente)

```
numfig = True
#language = 'ja'

extensions.append('sphinx.ext.todo')
extensions.append('sphinx.ext.autodoc')
#extensions.append('sphinx.ext.autosummary')
extensions.append('sphinx.ext.intersphinx')
extensions.append('sphinx.ext.mathjax')
extensions.append('sphinx.ext.viewcode')
extensions.append('sphinx.ext.graphviz')

autosummary_generate = True
html_theme = 'default'
#source_suffix = ['.rst', '.txt']
```

4.5 I Builder

Questi sono i builder nativi di Sphinx. Si possono aggiungere altri builder tramite *extensions*.

Il "nome" del builder deve essere fornito alle opzioni della riga di comando **-M** o **-b** di **sphinx-build** per selezionare un builder.

I builder più comuni sono:

html

Build di pagine HTML. Questo è il builder di default.

dirhtml

Costruisce pagine HTML, ma con un'unica directory per ogni documento. Abbellisce gli URL (non `.html`) se serviti da un webserver.

singlehtml

Costruisce un singolo HTML con l'intero contenuto.

htmlhelp, qthelp, devhelp, epub

Crea file HTML con informazioni aggiuntive per creare una raccolta di documentazione in uno di questi formati.

applehelp

Costruisce un Apple Help Book. Richiede **hiutil** e **codesign**, che non sono Open Source e attualmente sono disponibili solo su Mac OS X 10.6 e versioni successive.

latex

Crea sorgenti LaTeX compilabili in un documento PDF utilizzando **pdflatex**.

man

Costruisce pagine manual nel formato groff per sistemi UNIX.

texinfo

Costruisce file Texinfo che possono essere elaborati in file Info usando **makeinfo**.

text

Crea file di semplice testo.

gettext

Crea cataloghi di messaggi in stile gettext (file .pot).

doctest

Esegue tutti i doctest nella documentazione, se è abilitata l'estensione *doctest*.

linkcheck

Controlla l'integrità di tutti i link esterni.

xml

Crea file XML nativi di Docutils.

pseudoxml

Crea file "pseudo-XML" compatti e ben stampati che mostrino la struttura interna degli alberi di documenti intermedi.

class sphinx.builders.html.StandaloneHTMLBuilder

Questo è il builder HTML standard. Il suo output è una directory con file HTML, completa di fogli di stile e opzionalmente dei sorgenti reST. Ci sono alcuni valori di configurazione che personalizzano l'output di questo builder, vedere il capitolo *Opzioni per l'output HTML* per i dettagli.

name = 'html'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'html'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: list²¹⁶[str²¹⁷] = ['image/svg+xml', 'image/png', 'image/gif', 'image/jpeg']

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

class sphinx.builders.dirhtml.DirectoryHTMLBuilder

Questa è una sottoclasse del builder HTML standard. Il suo output è una directory con file HTML, dove ciascun file è chiamato `index.html` e inserito in una sottodirectory con lo stesso nome della sua pagina. Ad esempio, il documento `markup/rest.rst` non darà come risultato un file di output `markup/rest.html`, ma `markup/rest/index.html`. Quando si generano link tra pagine, `index.html` viene omissso, in modo che l'URL assomigli a `markup/rest/`.

name = 'dirhtml'

Il nome del builder, per l'opzione della riga di comando -b.

²¹⁶ <https://docs.python.org/3/library/stdtypes.html#list>

²¹⁷ <https://docs.python.org/3/library/stdtypes.html#str>

```
format = 'html'
```

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

```
supported_image_types: list218[str219] = ['image/svg+xml',  
'image/png', 'image/gif', 'image/jpeg']
```

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 0.6.

class sphinx.builders.singlehtml.SingleFileHTMLBuilder

Questo è un builder HTML che combina l'intero progetto in un unico file di output. (Ovviamente questo funziona solo con progetti più piccoli). Il file ha il nome del documento root. Non verrà generato alcun indice.

```
name = 'singlehtml'
```

Il nome del builder, per l'opzione della riga di comando -b.

```
format = 'html'
```

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

```
supported_image_types: list220[str221] = ['image/svg+xml',  
'image/png', 'image/gif', 'image/jpeg']
```

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 1.0.

class sphinxcontrib.htmlhelp.HTMLHelpBuilder

Questo builder produce lo stesso output del builder HTML, ma genera anche file di supporto della HTML Help che consentono a Microsoft HTML Help Workshop di compilarli in un file CHM.

```
name = 'htmlhelp'
```

Il nome del builder, per l'opzione della riga di comando -b.

```
format = 'html'
```

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

```
supported_image_types: list222[str223] = ['image/png', 'image/gif',  
'image/jpeg']
```

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

²¹⁸ <https://docs.python.org/3/library/stdtypes.html#list>

²¹⁹ <https://docs.python.org/3/library/stdtypes.html#str>

²²⁰ <https://docs.python.org/3/library/stdtypes.html#list>

²²¹ <https://docs.python.org/3/library/stdtypes.html#str>

²²² <https://docs.python.org/3/library/stdtypes.html#list>

²²³ <https://docs.python.org/3/library/stdtypes.html#str>

class sphinxcontrib.qthelp.QtHelpBuilder

Questo builder produce lo stesso output del builder HTML autonomo, ma genera anche file di supporto della raccolta [Qt help](https://doc.qt.io/qt-4.8/qthelp-framework.html)²²⁴ che consentono al generatore di raccolte Qt di compilarli.

Cambiato nella versione 2.0: Spostato in sphinxcontrib.qthelp dal pacchetto sphinx.builders.

name = 'qthelp'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'html'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: [list](https://docs.python.org/3/library/stdtypes.html#list)²²⁵ [[str](https://docs.python.org/3/library/stdtypes.html#str)²²⁶] = ['image/svg+xml', 'image/png', 'image/gif', 'image/jpeg']

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

class sphinxcontrib.applehelp.AppleHelpBuilder

Questo builder produce un Apple Help Book basato sullo stesso output del builder HTML.

Se la directory sorgente contiene cartelle .lproj, quella corrispondente alla lingua selezionata avrà il suo contenuto unito all'output generato. Queste cartelle verranno ignorate da tutti gli altri tipi di documentazione.

Per generare un valido help book, questo builder richiede il tool a riga di comando **hiutil**, disponibile solo su Mac OS X 10.6 e versioni successive. Si può disabilitare il passo di indicizzazione impostando [applehelp_disable_external_tools](#) a True, nel qual caso l'output non sarà valido finché **hiutil** non sarà stato eseguito su tutte le cartelle .lproj all'interno del pacchetto.

name = 'applehelp'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'html'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: [list](https://docs.python.org/3/library/stdtypes.html#list)²²⁷ [[str](https://docs.python.org/3/library/stdtypes.html#str)²²⁸] = ['image/png', 'image/gif', 'image/jpeg', 'image/tiff', 'image/jp2', 'image/svg+xml']

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 1.3.

Cambiato nella versione 2.0: Spostato in sphinxcontrib.applehelp dal pacchetto sphinx.builders.

²²⁴ <https://doc.qt.io/qt-4.8/qthelp-framework.html>

²²⁵ <https://docs.python.org/3/library/stdtypes.html#list>

²²⁶ <https://docs.python.org/3/library/stdtypes.html#str>

²²⁷ <https://docs.python.org/3/library/stdtypes.html#list>

²²⁸ <https://docs.python.org/3/library/stdtypes.html#str>

class sphinxcontrib.devhelp.DevhelpBuilder

Questo builder produce lo stesso output del builder HTML, ma genera anche il file di supporto [GNOME Devhelp](#)²²⁹ che consente al lettore GNOME Devhelp di visualizzarli.

name = 'devhelp'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'html'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: `list`²³⁰`[str`²³¹`] = ['image/png', 'image/gif', 'image/jpeg']`

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Cambiato nella versione 2.0: Spostato in sphinxcontrib.devhelp dal pacchetto sphinx.builders.

class sphinx.builders.epub3.Epub3Builder

Questo builder produce lo stesso output del builder HTML, ma genera anche un file *epub* per i lettori di ebook. Vedere [Info Epub](#) per i dettagli a riguardo. Per la definizione del formato epub, consultare <https://idpf.org/epub> o <https://en.wikipedia.org/wiki/EPUB>. Il builder crea file *EPUB 3*.

name = 'epub'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'html'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: `list`²³²`[str`²³³`] = ['image/svg+xml', 'image/png', 'image/gif', 'image/jpeg']`

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 1.4.

Cambiato nella versione 1.5: A partire da Sphinx 1.5, il builder epub3 viene utilizzato come builder epub di default.

class sphinx.builders.latex.LaTeXBuilder

Questo builder produce file sorgente LaTeX nella directory di output. Le build PDF effettive avvengono all'interno di questa directory di output e devono essere attivate in un secondo passaggio. Questo può essere fatto tramite **make all-pdf**. Per combinare i due passaggi in uno solo, si usa *sphinx-build -M* (cioè -M latexpdf non -b latexpdf) oppure **make latexpdf** nella root del progetto.

²²⁹ <https://wiki.gnome.org/Apps/Devhelp>

²³⁰ <https://docs.python.org/3/library/stdtypes.html#list>

²³¹ <https://docs.python.org/3/library/stdtypes.html#str>

²³² <https://docs.python.org/3/library/stdtypes.html#list>

²³³ <https://docs.python.org/3/library/stdtypes.html#str>

Consultare [latex_documents](#) e il capitolo *Opzioni per l'output LaTeX* per le opzioni disponibili.

Le build PDF richiedono un'installazione di LaTeX sufficientemente completa. Il test è attualmente (dalla versione 5.3.0) eseguito su Ubuntu 22.04 LTS, la cui distribuzione LaTeX corrisponde a TeXLive 2021 upstream a partire dal 04/02/2022, ma le build PDF si possono eseguire con successo su installazioni LaTeX molto più vecchie.

In ogni caso, su Ubuntu ad esempio, devono essere tutti presenti i seguenti pacchetti:

- `texlive-latex-recommended`
- `texlive-fonts-recommended`
- `tex-gyre` (se [latex_engine](#) è lasciato col default)
- `texlive-latex-extra`
- `latexmk`

Cambiato nella versione 4.0.0: I font TeX Gyre ora sono richiesti per il motore 'pdf`latex`' (il default).

In alcune circostanze sono necessari pacchetti aggiuntivi:

- `texlive-lang-cyrillic` per il cirillico (e anche `cm-super` se si utilizzano i font di default),
- `texlive-lang-greek` per il greco (e anche `cm-super` se si utilizzano i font di default),
- `texlive-xetex` se [latex_engine](#) è 'xetex',
- `texlive-luatex` se [latex_engine](#) è 'lua`latex`',
- `fonts-freefont-otf` se [latex_engine](#) è 'xetex' o 'lua`latex`'.

Nota: Dalla versione 1.6, `make latexpdf` utilizza su GNU/Linux e macOS **latexmk**, poiché si assicura che il numero di esecuzioni necessarie venga eseguito automaticamente. Su Windows le build PDF eseguono un numero fisso di esecuzioni LaTeX (tre, poi `makeindex`, poi altre due).

È possibile passare alle opzioni `latexmk` tramite la variabile Makefile `LATEXMKOPTS`. Per esempio:

```
make latexpdf LATEXMKOPTS="-silent"
```

riduce al minimo l'output della console.

Inoltre, se `latexmk` è alla versione 4.52b o successiva (gennaio 2017) `LATEXMKOPTS="-xetex"` accelera la creazione di PDF tramite XeLaTeX in caso di numerose inclusioni grafiche.

Per passare le opzioni direttamente al binario `(pdf|xe|lua)latex`, si usa la variabile `LATEXOPTS`, ad esempio:

```
make latexpdf LATEXOPTS="--halt-on-error"
```

name = 'latex'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'latex'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: list²³⁴[str²³⁵] = ['application/pdf', 'image/png', 'image/jpeg']

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Notare che [rinohtype²³⁶](#) fornisce un builder PDF diretto. Il nome del builder è `rinoh`. Fare riferimento al [rinohtype manual²³⁷](#) per i dettagli.

class sphinx.builders.text.TextBuilder

Questo builder produce un file di testo per ogni file reST: è quasi uguale al sorgente reST, ma con gran parte del markup rimossi per una migliore leggibilità.

name = 'text'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'text'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: list²³⁸[str²³⁹] = []

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 0.4.

class sphinx.builders.manpage.ManualPageBuilder

Questo builder produce pagine di manuale nel formato groff. È necessario specificare quali documenti devono essere inclusi in quali pagine di manuale tramite il valore della configurazione di `man_pages`.

name = 'man'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'man'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

²³⁴ <https://docs.python.org/3/library/stdtypes.html#list>

²³⁵ <https://docs.python.org/3/library/stdtypes.html#str>

²³⁶ <https://github.com/brechtm/rinohtype>

²³⁷ <https://www.mos6581.org/rinohtype/master/quickstart.html#sphinx-builder>

²³⁸ <https://docs.python.org/3/library/stdtypes.html#list>

²³⁹ <https://docs.python.org/3/library/stdtypes.html#str>

supported_image_types: `list240[str241] = []`

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 1.0.

class sphinx.builders.texinfo.TexinfoBuilder

Questo builder produce file Texinfo che possono essere elaborati in file Info dal programma **makeinfo**. Si devono specificare quali documenti includere in quali file Texinfo tramite il valore della configurazione `texinfo_documents`.

Il formato Info è la base del sistema di help on-line utilizzato da GNU Emacs e dal programma basato su terminale **info**. Vedere *Info su Texinfo* per ulteriori dettagli. Il formato Texinfo è il sistema di documentazione ufficiale utilizzato dal progetto GNU. Maggiori informazioni su Texinfo possono essere trovate su <https://www.gnu.org/software/texinfo/>.

name = 'texinfo'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'texinfo'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: `list242[str243] = ['image/png', 'image/jpeg', 'image/gif']`

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 1.1.

class sphinxcontrib.serializinghtml.SerializingHTMLBuilder

Questo builder utilizza un modulo che implementa l'API di serializzazione Python (pickle, simplejson, phpserialize ed altri) per eseguire il dump della documentazione HTML generata. Il builder pickle ne è una sottoclasse.

Una sottoclasse concreta di questo builder nel formato `PHP serialization244` potrebbe assomigliare a questa:

```
import phpserialize

class PHPSerializedBuilder(SerializingHTMLBuilder):
    name = 'phpserialized'
    implementation = phpserialize
    out_suffix = '.file.phpdump'
    globalcontext_filename = 'globalcontext.phpdump'
    searchindex_filename = 'searchindex.phpdump'
```

²⁴⁰ <https://docs.python.org/3/library/stdtypes.html#list>

²⁴¹ <https://docs.python.org/3/library/stdtypes.html#str>

²⁴² <https://docs.python.org/3/library/stdtypes.html#list>

²⁴³ <https://docs.python.org/3/library/stdtypes.html#str>

implementation

Un modulo che implementa le funzioni `dump()`, `load()`, `dumps()` e `loads()` conformi alle funzioni con gli stessi nomi del modulo `pickle`. I moduli noti che implementano questa interfaccia sono `simplejson`, `phpserialize`, `plistlib`, e altri.

out_suffix

Il suffisso per tutti i file normali.

globalcontext_filename

Il nome del file che contiene il "global context". Questo è un dict con alcuni valori di configurazione generali come il nome del progetto.

searchindex_filename

Il nome file per l'indice di ricerca generato da Sphinx.

Vedere *Dettagli del builder di serializzazione* per i dettagli sul formato di output.

Added in version 0.5.

class sphinxcontrib.serializinghtml.PickleHTMLBuilder

Questo builder produce una directory con file pickle contenenti principalmente frammenti HTML e informazioni TOC, per l'uso di un'applicazione web (o strumento di postelaborazione personalizzato) che non utilizza i template HTML standard.

Vedere *Dettagli del builder di serializzazione* per i dettagli sul formato di output.

name = 'pickle'

Il nome del builder, per l'opzione della riga di comando `-b`.

Il vecchio nome "web" funziona ancora.

format = 'html'

Il formato di output del builder o " se non viene prodotto alcun output del documento.

supported_image_types: `list245[str246] = ['image/svg+xml', 'image/png', 'image/gif', 'image/jpeg']`

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Il suffisso del file è `.fpickle`. Il contesto globale è chiamato `globalcontext.pickle`, l'indice di ricerca `searchindex.pickle`.

class sphinxcontrib.serializinghtml.JSONHTMLBuilder

Questo builder produce una directory con file JSON contenenti principalmente frammenti HTML e informazioni TOC, per l'utilizzo di un'applicazione Web (o strumento di postelaborazione personalizzato) che non utilizza i template HTML standard.

Vedere *Dettagli del builder di serializzazione* per i dettagli sul formato di output.

²⁴⁴ <https://pypi.org/project/phpserialize/>

²⁴⁵ <https://docs.python.org/3/library/stdtypes.html#list>

²⁴⁶ <https://docs.python.org/3/library/stdtypes.html#str>

name = 'json'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'html'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: `list`²⁴⁷[`str`²⁴⁸] = ['image/svg+xml',
'image/png', 'image/gif', 'image/jpeg']

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Il suffisso del file è .fjson. Il contesto globale è chiamato `globalcontext.json`, l'indice di ricerca `searchindex.json`.

Added in version 0.5.

class sphinx.builders.gettext.MessageCatalogBuilder

Questo builder produce cataloghi di messaggi in stile gettext. Ogni file o sottodirectory di livello superiore crea un singolo template di catalogo .pot.

Consultare la documentazione su *Internazionalizzazione* per ulteriori riferimenti.

name = 'gettext'

Il nome del builder, per l'opzione della riga di comando -b.

format = ''

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: `list`²⁴⁹[`str`²⁵⁰] = []

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 1.1.

class sphinx.builders.changes.ChangesBuilder

Questo builder produce una panoramica [overview] HTML di tutte le direttive *versionadded*, *versionchanged*, *deprecated* e *versionremoved* per la *version* corrente. Ciò è utile, ad esempio, per generare un file di log delle modifiche.

name = 'changes'

Il nome del builder, per l'opzione della riga di comando -b.

format = ''

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: `list`²⁵¹[`str`²⁵²] = []

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

²⁴⁷ <https://docs.python.org/3/library/stdtypes.html#list>

²⁴⁸ <https://docs.python.org/3/library/stdtypes.html#str>

²⁴⁹ <https://docs.python.org/3/library/stdtypes.html#list>

²⁵⁰ <https://docs.python.org/3/library/stdtypes.html#str>

class sphinx.builders.dummy.DummyBuilder

Questo builder non produce alcun output. L'input viene solo analizzato e controllato per verificarne la coerenza. Questo è utile per debug [linting].

name = 'dummy'

Il nome del builder, per l'opzione della riga di comando -b.

supported_image_types: `list253[str254]` = []

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 1.4.

class sphinx.builders.linkcheck.CheckExternalLinksBuilder

Questo builder scansiona tutti i documenti alla ricerca di link esterni, prova ad aprirli con delle requests e scrive una panoramica di quali sono danneggiati e reindirizzati all'output standard e a `output.txt` nella directory di output.

name = 'linkcheck'

Il nome del builder, per l'opzione della riga di comando -b.

format = ''

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: `list255[str256]` = []

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Cambiato nella versione 1.5: A partire da Sphinx 1.5, il linkcheck builder utilizza il modulo request.

Cambiato nella versione 3.4: Il builder linkcheck ritenta i collegamenti quando i server applicano limiti di velocità.

class sphinx.builders.xml.XMLBuilder

Questo builder produce file XML nativi di Docutils. L'output può essere trasformato con strumenti XML standard come i processori XSLT in formati finali arbitrari.

name = 'xml'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'xml'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: `list257[str258]` = []

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

²⁵¹ <https://docs.python.org/3/library/stdtypes.html#list>

²⁵² <https://docs.python.org/3/library/stdtypes.html#str>

²⁵³ <https://docs.python.org/3/library/stdtypes.html#list>

²⁵⁴ <https://docs.python.org/3/library/stdtypes.html#str>

²⁵⁵ <https://docs.python.org/3/library/stdtypes.html#list>

²⁵⁶ <https://docs.python.org/3/library/stdtypes.html#str>

Added in version 1.2.

class sphinx.builders.xml.PseudoXMLBuilder

Questo builder viene utilizzato per il debug della pipeline "Reader to Transform to Writer" di Sphinx/Docutils. Produce file "pseudo-XML" compatti e formattati in cui l'annidamento è indicato dal rientro (senza tag finali). Vengono scritti gli attributi esterni per tutti gli elementi e vengono forniti anche gli attributi interni per eventuali elementi "in sospeso" rimanenti.

name = 'pseudoxml'

Il nome del builder, per l'opzione della riga di comando -b.

format = 'pseudoxml'

Il formato di output del builder o "" se non viene prodotto alcun output del documento.

supported_image_types: `list259[str260] = []`

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

Added in version 1.2.

Le estensioni Sphinx native che offrono altri builder sono:

- *doctest*
- *coverage*

Dettagli del builder di serializzazione

Tutti i builder di serializzazione generano un file per ogni file sorgente e alcuni file speciali. Copiano anche i file sorgente reST nella directory `_sources` nella directory di output.

La *PickleHTMLBuilder* è una sottoclasse che implementa l'interfaccia di serializzazione pickle.

I file per i file sorgenti hanno l'estensione *out_suffix* e sono organizzati in directory proprio come i file sorgenti. De-serializzano in un dictionary (o in una struttura simile a un dizionario) con queste chiavi:

body

Il "body" HTML (ovvero, come appare l'HTML del file sorgente), come reso dal traduttore HTML.

title

Il titolo del documento, in formato HTML (può contenere markup).

toc

Il sommario del file [TOC], reso come HTML ``.

²⁵⁷ <https://docs.python.org/3/library/stdtypes.html#list>

²⁵⁸ <https://docs.python.org/3/library/stdtypes.html#str>

²⁵⁹ <https://docs.python.org/3/library/stdtypes.html#list>

²⁶⁰ <https://docs.python.org/3/library/stdtypes.html#str>

display_toc

Un booleano che è True se il toc contiene più di una voce.

current_page_name

Il nome del documento del file corrente.

parents, prev and next

Informazioni sui capitoli correlati nell'albero del sommario [TOC]. Ogni relazione è un dizionario con le chiavi `link` (HREF per la relazione) e `title` (titolo del documento correlato, in formato HTML). `parents` è un elenco di relazioni, mentre `prev` e `next` sono una singola relazione.

sourcename

Il nome del file sorgente in `_sources`.

I file speciali si trovano nella directory di output root. Sono:

SerializingHTMLBuilder.globalcontext_filename

Un dict pickled con queste chiavi:

project, copyright, release, version

Gli stessi valori forniti nel file di configurazione.

style

html_style.

last_updated

Data dell'ultima build.

builder

Nome del builder utilizzato, nel caso dei pickle è sempre 'pickle'.

titles

Un dizionario di tutti i titoli dei documenti, come stringhe HTML.

SerializingHTMLBuilder.searchindex_filename

Un indice utilizzabile per la ricerca nella documentazione. È un elenco pickle con queste voci:

- Un elenco di nomi di documenti indicizzati.
- Un elenco di titoli di documenti, come stringhe HTML, nello stesso ordine del primo elenco.
- Un dict che mappa le radici delle parole (elaborate da uno stemmer in lingua inglese) a un elenco di numeri interi, che sono indici nel primo elenco.

environment.pickle

L'ambiente di build. Si tratta sempre di un file pickle, indipendente dal builder e una copia dell'ambiente utilizzato all'avvio del builder.

Da fare: Documentare i membri comuni.

A differenza degli altri file pickle, questo file pickle richiede che il pacchetto `sphinx` sia disponibile al momento dell'unpickling.

4.6 Domini

Added in version 1.0.

Originariamente Sphinx è stato concepito per un unico progetto, la documentazione del linguaggio Python. Poco dopo, fu reso disponibile a tutti come strumento di documentazione, ma la documentazione dei moduli Python rimase profondamente integrata: le direttive più fondamentali, come `function`, furono progettate per oggetti Python. Poiché Sphinx è diventato piuttosto popolare, si è sviluppato l'interesse nell'utilizzarlo per molti altri scopi: progetti C/C++, JavaScript o persino markup reStructuredText (come in questa documentazione).

Sebbene ciò sia sempre stato possibile, ora è molto più semplice supportare facilmente la documentazione di progetti utilizzando diversi linguaggi di programmazione o anche quelli non supportati dalla distribuzione principale di Sphinx, fornendo un **dominio** per ciascuno di questi scopi.

Un dominio è una raccolta di markup (*directive* e *role* reStructuredText) per descrivere e creare link a relativi *object*, ad es. elementi di un linguaggio di programmazione. I nomi delle direttive e dei ruoli in un dominio hanno nomi come `domain:name`, ad es. `py:function`. I domini possono anche fornire indici personalizzati (come il Python Module Index).

Avere domini significa che non ci sono problemi di nomenclatura quando si vuole fare riferimento a una serie di documentazione, ad es. classi C++ e Python. Significa anche che le estensioni che supportano la documentazione di linguaggi completamente nuovi sono molto più facili da scrivere.

Questa sezione descrive cosa forniscono i domini inclusi in Sphinx. Anche l'API del dominio è documentata nella sezione *API di dominio*.

Markup di base

La maggior parte dei domini fornisce una serie di *direttive di descrizione dell'oggetto*, utilizzate per descrivere oggetti specifici forniti dai moduli. Ogni direttiva richiede una o più 'signature' (firme) per fornire informazioni di base su ciò che viene descritto e il contenuto dovrebbe essere la descrizione.

Un dominio in genere manterrà un indice interno di tutte le entità per facilitare i riferimenti incrociati [cross-reference]. In genere aggiungerà anche voci nell'indice generale. Per evitare l'aggiunta di una voce nell'indice, si può fornire il flag, opzione di direttiva, `:no-index-entry:`. Per escludere la descrizione dell'oggetto dal sommario, si può fornire il flag, opzione di direttiva, `:no-contents-entry:`. Per comporre la descrizione di un oggetto, senza nemmeno renderla disponibile per i riferimenti incrociati, si può fornire il flag, opzione di direttiva, `:no-index:` (che implica `:no-index-entry:`). Per non scrivere nulla, si può fornire il flag, opzione di direttiva, `:no-typesetting:`. Questo può essere utilizzato, ad esempio, per creare solo una destinazione [target] e una voce di indice per un riferimento

futuro. Da tener presente, però, che non tutte le direttive possono supportare queste opzioni in qualsiasi dominio.

Added in version 3.2: L'opzione della direttiva `noindexentry` nei domini Python, C, C++ e Javascript.

Added in version 5.2.3: L'opzione direttiva `:nocontentsentry:` nei domini Python, C, C++, Javascript e reStructuredText.

Added in version 7.2: L'opzione direttiva `no-typesetting` nei domini Python, C, C++, Javascript e reStructuredText.

Cambiato nella versione 7.2:

- L'opzione direttiva `:noindex:` è stata rinominata `:no-index:`.
- L'opzione direttiva `:noindexentry:` è stata rinominata `:no-index-entry:`.
- L'opzione direttiva `:nocontentsentry:` è stata rinominata `:no-contents-entry:`.

I nomi precedenti vengono mantenuti come alias, ma saranno deprecati e rimossi in una versione futura di Sphinx.

Un esempio che utilizza una direttiva del dominio Python:

```
.. py:function:: spam(eggs)
               ham(eggs)
```

Spam or ham the foo.

Descrive le due funzioni Python `spam` e `ham`. (Notare che quando le firme diventano troppo lunghe, si possono interrompere aggiungendo una barra rovesciata alle righe che continuano nella riga successiva. Esempio:

```
.. py:function:: filterwarnings(action, message='', category=Warning, \
                               module='', lineno=0, append=False)
:~no-index:
```

(Questo esempio mostra anche come utilizzare il flag `:no-index:`).

I domini forniscono anche ruoli che rimandano a queste descrizioni di oggetti. Ad esempio, per collegarsi a una delle funzioni descritte nell'esempio sopra, si potrebbe dire:

```
The function :py:func:`spam` does a similar thing.
```

Come si vede, sia i nomi delle direttive che quelli dei ruoli contengono il nome del dominio e quello della direttiva.

L'opzione direttiva `:no-typesetting:` si può usare per creare una destinazione `[target]` (e una voce di indice) a cui possono successivamente fare riferimento i ruoli forniti dal dominio. Ciò è particolarmente utile per la "literate programming":

```
.. py:function:: spam(eggs)
:~no-typesetting:
```

(continues on next page)

(continua dalla pagina precedente)

```
.. code::
```

```
def spam(eggs):
    pass
```

The function `:py:func:`spam`` does nothing.

Dominio di Default

Per la documentazione che descrive oggetti appartenenti ad un solo dominio, gli autori non dovranno dichiararne nuovamente il nome ad ogni direttiva, ruolo, ecc... dopo aver specificato un valore di default. Questo può essere fatto tramite il valore di configurazione `primary_domain` o tramite questa direttiva:

```
.. default-domain:: name
```

Seleziona un nuovo dominio di default. Anche se `primary_domain` seleziona un valore di default globale, questo ha effetto solo all'interno dello stesso file.

Se non viene selezionato nessun altro dominio di default, quello Python (chiamato `py`) è quello di default, principalmente per compatibilità con la documentazione scritta per le versioni precedenti di Sphinx.

Direttive e ruoli che appartengono al dominio di default possono essere menzionati senza indicare il nome del dominio, ad esempio:

```
.. function:: pyfunc()
```

Describes a Python function.

Reference to `:func:`pyfunc``.

Sintassi dei riferimenti incrociati

Per i ruoli dei riferimenti incrociati forniti dai domini, esistono le stesse funzionalità di quelli generali. Vedere *Sintassi dei riferimenti incrociati*.

In breve:

- Si può fornire un titolo esplicito e un target di riferimento, come nei link ipertestuali reST diretti: `:role:`title <target>`` si riferirà a *target*, ma il testo del link sarà *title*.
- Se si aggiunge al contenuto il prefisso `!`, non verrà creato alcun riferimento/link ipertestuale.
- Se al contenuto si aggiunge il prefisso `~`, il testo del link sarà solo l'ultimo componente del target. Per esempio, `:py:meth:`~Queue.Queue.get`` farà riferimento a `Queue.Queue.get` ma visualizzerà solo `get` come testo del link.

Domini nativi

I seguenti domini sono inclusi in Sphinx:

Il Dominio Standard

Added in version 1.0.

Il cosiddetto dominio "standard" raccoglie tutti i markup che non hanno un dominio proprio. Le sue direttive e ruoli non hanno come prefisso un nome di dominio.

Il dominio standard è anche il luogo in cui vengono posizionate le descrizioni degli oggetti custom, aggiunte utilizzando l'API `add_object_type()`.

Esiste una serie di direttive che consentono di documentare i programmi a riga di comando:

.. option:: name args, name args, ...

Descrive un argomento o un'opzione della riga di comando. I nomi degli argomenti delle opzioni devono essere racchiusi tra parentesi angolari. Esempi:

```
.. option:: dest_dir

    Destination directory.

.. option:: -m <module>, --module <module>

    Run a module as a script.
```

La direttiva creerà destinazioni di riferimenti incrociati per le opzioni specificate, referenziabili da `option` (nel caso dell'esempio, si userebbe qualcosa come `:option:`dest_dir`, :option:`-m` oppure :option:`--module`).`

Cambiato nella versione 5.3: Si può fare un riferimento incrociato includendo un valore di opzione: `:option:`--module=foobar`, ``:option:--module[=foobar]`` o :option:`--module foobar`.`

Si usa `option_emphasise_placeholders` per analizzare la "parte variabile" di un testo letterale (in modo simile al ruolo `samp`).

La direttiva `cmdoption` è un alias deprecato per la direttiva `option`.

.. envvar:: name

Descrive una variabile di ambiente che il codice o il programma documentato utilizza o definisce. Referenziabile da `envvar`.

.. program:: name

Come `py:currentmodule`, questa direttiva non produce alcun output. Serve però a notificare a Sphinx che tutte le seguenti direttive `option` documentano le opzioni per il programma chiamato *name*.

Se si usa `program`, si devono qualificare i riferimenti nei propri ruoli `option` con il nome del programma, quindi se si è nella seguente situazione:


```

.. program:: rm

.. option:: -r

    Work recursively.

.. program:: svn

.. option:: -r <revision>

    Specify the revision to work upon.

```

allora `:option:`rm -r`` si riferirebbe alla prima opzione, mentre `:option:`svn -r`` si riferirebbe alla seconda.

Se all'argomento viene passato `None`, la direttiva reimposterà il nome del programma corrente.

Il nome del programma può contenere spazi (nel caso in cui si desideri documentare sottocomandi come `svn add` e `svn commit` separatamente).

Added in version 0.5.

C'è anche una direttiva sulla descrizione dell'oggetto molto generica, che non è legata ad alcun dominio:

```

.. describe:: text
.. object:: text

```

Questa direttiva produce la stessa formattazione di quelle specifiche fornite dai domini, ma non crea voci di indice o target di riferimenti incrociati. Esempio:

```

.. describe:: PAPER

    You can set this variable to select a paper size.

```

Il Dominio C

Added in version 1.0.

Il dominio C (nome `c`) è adatto per la documentazione delle API C.

```

.. c:member:: declaration
.. c:var:: declaration

```

Descrive un membro o una variabile della struct C. Esempio di signature:

```

.. c:member:: PyObject *PyObject.tp_bases

```

La differenza tra le due direttive è solo estetica.

.. c:function:: function prototype

Descrive una funzione C. La firma deve essere quella del C, ad esempio:

```
.. c:function:: PyObject *PyType_GenericAlloc(PyTypeObject *type,
↳Py_ssize_t nitems)
```

Da notare che non è necessario eseguire l'escape con il backslash degli asterischi nella firma, poiché non viene analizzata da reST.

Nella descrizione di una funzione si possono utilizzare i seguenti campi info (vedere anche *Liste di campi info*).

- **param**, **parameter**, **arg**, **argument**: Descrizione di un parametro.
- **type**: Il tipo di un parametro, scritto come se fosse passato al ruolo **c:expr**.
- **returns**, **return**: Descrizione del valore restituito.
- **rtype**: Il tipo restituito, scritto come se fosse passato al ruolo **c:expr**.
- **retval**, **retvals**: Un'alternativa a **returns** per descrivere il risultato della funzione.

Added in version 4.3: Il tipo del campo **retval**.

Per esempio:

```
.. c:function:: PyObject *PyType_GenericAlloc(PyTypeObject *type,
↳Py_ssize_t nitems)

:param type: description of the first parameter.
:param nitems: description of the second parameter.
:returns: a result.
:retval NULL: under some conditions.
:retval NULL: under some other conditions as well.
```

appare come

`PyObject261 *PyType_GenericAlloc(PyTypeObject262 *type, Py_ssize_t263 nitems)`

No-contents-entry

No-index-entry

Parametri

- **type** -- Descrizione del primo parametro.
- **nitems** -- descrizione del secondo parametro.

Ritorna

un risultato.

Valori restituiti

- **NULL** -- in certe condizioni.
- **NULL** -- secondo anche altre condizioni.

:single-line-parameter-list: (no value)

Garantisce che i parametri della funzione verranno emessi su una singola linea logica, sovrascrivendo `c_maximum_signature_line_length` e `maximum_signature_line_length`.

Added in version 7.1.

.. c:macro:: name**.. c:macro:: name(arg list)**

Descrive una macro C, cioè una `#define` del linguaggio C, senza il testo sostitutivo.

Nella descrizione di una macro è possibile utilizzare gli stessi campi info della direttiva `c:function`.

Added in version 3.0: La variante dello stile della funzione.

:single-line-parameter-list: (no value)

Garantisce che i parametri della funzione verranno emessi su una singola linea logica, sovrascrivendo `c_maximum_signature_line_length` e `maximum_signature_line_length`.

Added in version 7.1.

.. c:struct:: name

Descrive una struct C.

Added in version 3.0.

.. c:union:: name

Descrive una union C.

Added in version 3.0.

.. c:enum:: name

Descrive una enum C.

Added in version 3.0.

.. c:enumerator:: name

Descrive un enumeratore C.

Added in version 3.0.

.. c:type:: typedef-like declaration**.. c:type:: name**

Descrive un tipo C, come typedef o l'alias per un tipo non specificato.

²⁶¹ <https://docs.python.org/3/c-api/structures.html#c.PyObject>

²⁶¹ <https://docs.python.org/3/c-api/structures.html#c.PyObject>

²⁶² <https://docs.python.org/3/c-api/type.html#c.PyTypeObject>

²⁶³ https://docs.python.org/3/c-api/intro.html#c.Py_ssize_t

Riferimenti incrociati a costrutti C

I seguenti ruoli creano riferimenti incrociati ai costrutti del linguaggio C se sono definiti nella documentazione:

`:c:member:`
`:c:data:`
`:c:var:`
`:c:func:`
`:c:macro:`
`:c:struct:`
`:c:union:`
`:c:enum:`
`:c:enumerator:`
`:c:type:`

Riferimento a una dichiarazione C, come definito sopra. Notare che `c:member`, `c:data` e `c:var` sono equivalenti.

Added in version 3.0: I ruoli var, struct, union, enum ed enumerator.

Entità Anonime

Il C supporta strutture, enum e union, tutte anonime. Ai fini della documentazione è necessario assegnare loro un nome che inizi con @, ad esempio, @42 o @data. Tali nomi possono essere utilizzati anche nei riferimenti incrociati, sebbene i simboli annidati verranno trovati anche se omessi. Il nome @... apparirà sempre come **[anonymous]** (possibilmente come link).

Esempio:

```
.. c:struct:: Data

    .. c:union:: @data

        .. c:var:: int a

        .. c:var:: double b

Explicit ref: :c:var:`Data.@data.a`. Short-hand ref: :c:var:`Data.a`.
```

Questo apparirà come:

```
struct Data

    union [anonymous]

        int a
```

double **b**

Esplicito ref: *Data*.*[anonymous]*.*a*. Abbreviato ref: *Data*.*a*.

Added in version 3.0.

Dichiarazioni di Alias

A volte può essere utile elencare le dichiarazioni altrove rispetto alla documentazione principale, ad esempio quando si crea una sinossi dell'interfaccia di una classe. A questo scopo può essere utilizzata la seguente direttiva.

.. c:alias:: name

Inserisce una o più dichiarazioni di alias. Ogni entità può essere specificata come nel ruolo `c:any`.

Per esempio:

```
.. c:var:: int data
.. c:function:: int f(double k)

.. c:alias:: data
           f
```

diventa

int **data**

int **f**(double k)

int *data*

int *f*(double k)

Added in version 3.2.

Opzioni

:maxdepth: int

Si inseriscono anche dichiarazioni nidificate, fino alla profondità totale data. Utilizzare 0 per profondità infinita e 1 solo per la dichiarazione menzionata. Il default è 1.

Added in version 3.3.

:noroot:

Salta le dichiarazioni menzionate e visualizza solo le quelle annidate. Richiede `maxdepth` 0 o almeno 2.

Added in version 3.5.

Espressioni e Tipi In Linea


:c:expr:

:c:texpr:

Inserisce un'espressione C++ o un tipo sia come codice inline (`cpp:expr`) che come testo inline (`cpp:texpr`). Per esempio:

```
.. c:var:: int a = 42
```

```
.. c:function:: int f(int i)
```

An expression: `:c:expr:`a * f(a)`` (or as text: `:c:texpr:`a *  f(a)``).

A type: `:c:expr:`const Data*``
(or as text `:c:texpr:`const Data*``).

apparirà in questo modo:

```
int a = 42
```

```
int f(int i)
```

Un'espressione: `a * f(a)` (or as text: `a * f(a)`).

Un tipo: `const Data*` (o come testo `const Data*`).

Added in version 3.0.

Lo Spazio dei Nomi

Added in version 3.1.

Il linguaggio C di per sé non supporta lo spazio dei nomi [namespace], ma a volte può essere utile emularlo nella documentazione, ad esempio per mostrare dichiarazioni alternative. La funzionalità può essere utilizzata anche per documentare i membri di strutture/unioni/enumerazioni separate dalla loro dichiarazione genitore.

Lo scope corrente può essere modificato utilizzando tre direttive sullo spazio dei nomi. Gestiscono dichiarazioni di stack in cui `c:namespace` reimposta lo stack e modifica un determinato scope.

La direttiva `c:namespace-push` cambia lo scope in uno più interno di quello corrente.

La direttiva `c:namespace-pop` annulla la direttiva `c:namespace-push`.

.. **c:namespace::** scope specification

Modifica lo scope corrente per gli oggetti successivi nell'ambito specificato e reimposta lo stack delle direttive dello spazio dei nomi. Notare che gli scope nidificati possono essere specificati separandoli con un punto, ad esempio:

```
.. c:namespace:: Namespace1.Namespace2.SomeStruct.AnInnerStruct
```

Tutti gli oggetti successivi verranno definiti come se il loro nome fosse dichiarato antepo-
nendo lo scope. I successivi riferimenti incrociati verranno ricercati a partire dallo scope
attuale.

Utilizzando NULL o \emptyset come scope si passerà allo scope globale.

.. c:namespace-push:: scope specification

Modifica lo scope relativamente a quello corrente. Per esempio, dopo:

```
.. c:namespace:: A.B
.. c:namespace-push:: C.D
```

lo scope corrente sarà A.B.C.D.

.. c:namespace-pop::

Annulla la precedente direttiva c:namespace-push (*non* il semplice 'pop' di uno scope).
Per esempio, dopo:

```
.. c:namespace:: A.B
.. c:namespace-push:: C.D
.. c:namespace-pop::
```

lo scope attuale sarà A.B (*non* A.B.C).

Se non è stata utilizzata alcuna direttiva c:namespace-push precedente, ma solo una
direttiva c:namespace, l'ambito corrente verrà reimpostato sullo scope globale. Cioè,
.. c:namespace:: A.B è equivalente a:

```
.. c:namespace:: NULL
.. c:namespace-push:: A.B
```

Variabili di Configurazione

Vedere *Opzioni per il dominio C*.

Il Dominio C++

Added in version 1.0.

Il dominio C++ (nome **cpp**) supporta la documentazione di progetti C++.

Direttive per gli Entità Dichiarative [Declaring Entities]

Sono disponibili le seguenti direttive. Tutte le dichiarazioni possono iniziare con un'istruzione di visibilità (`public`, `private` o `protected`).

.. **cpp:class::** class specifier

.. **cpp:struct::** class specifier

Descrive una classe/struttura, possibilmente specificando l'ereditarietà, ad esempio:

```
.. cpp:class:: MyClass : public MyBase, MyOtherBase
```

La differenza tra *cpp:class* e *cpp:struct* è solo estetica: il prefisso reso nell'output e lo specificatore mostrato nell'indice.

La classe può essere dichiarata direttamente all'interno di uno scope annidato, ad esempio:

```
.. cpp:class:: OuterScope::MyClass : public MyBase, MyOtherBase
```

Una class template si può dichiarare:

```
.. cpp:class:: template<typename T, std::size_t N> std::array
```

o con un'interruzione di riga:

```
.. cpp:class:: template<typename T, std::size_t N> \  
    std::array
```

È possibile dichiarare specializzazioni di template complete e parziali:

```
.. cpp:class:: template<> \  
    std::array<bool, 256>  
  
.. cpp:class:: template<typename T> \  
    std::array<T, 42>
```

Added in version 2.0: La direttiva *cpp:struct*.

.. cpp:function:: (member) function prototype

Descrive una funzione o una funzione membro, ad esempio:

```
.. cpp:function:: bool myMethod(int arg1, std::string arg2)

    A function with parameters and types.

.. cpp:function:: bool myMethod(int, double)

    A function with unnamed parameters.

.. cpp:function:: const T &MyClass::operator[](std::size_t i) const

    An overload for the indexing operator.

.. cpp:function:: operator bool() const

    A casting operator.

.. cpp:function:: constexpr void foo(std::string &bar[2]) noexcept

    A constexpr function.

.. cpp:function:: MyClass::MyClass(const MyClass&) = default

    A copy constructor with default implementation.
```

Si possono descrivere anche le funzioni template:

```
.. cpp:function:: template<typename U> \
    void print(U &&u)
```

e le specializzazioni della funzione template:

```
.. cpp:function:: template<> \
    void print(int i)
```

:single-line-parameter-list: (no value)

Garantisce che i parametri della funzione verranno emessi su una singola linea logica, sovrascrivendo `cpp_maximum_signature_line_length` e `maximum_signature_line_length`.

Added in version 7.1.

.. cpp:member:: (member) variable declaration

.. cpp:var:: (member) variable declaration

Descrive una variabile o una variabile membro, ad esempio:

```
.. cpp:member:: std::string MyClass::myMember  
  
.. cpp:var:: std::string MyClass::myOtherMember[N][M]  
  
.. cpp:member:: int a = 42
```

Si possono descrivere anche le variabili template:

```
.. cpp:member:: template<class T> \  
    constexpr T pi = T(3.1415926535897932385)
```

- .. **cpp:type**:: typedef declaration
- .. **cpp:type**:: name
- .. **cpp:type**:: type alias declaration

Descrive un tipo come in una dichiarazione typedef, una dichiarazione di alias di tipo o semplicemente il nome di un tipo non specificato, ad esempio:

```
.. cpp:type:: std::vector<int> MyList  
  
    A typedef-like declaration of a type.  
  
.. cpp:type:: MyContainer::const_iterator  
  
    Declaration of a type alias with unspecified type.  
  
.. cpp:type:: MyType = std::unordered_map<int, std::string>  
  
    Declaration of a type alias.
```

Un alias di tipo si può anche rendere template:

```
.. cpp:type:: template<typename T> \  
    MyContainer = std::vector<T>
```

Gli esempi sono resi come segue.

```
typedef std::vector<int> MyList
```

Una dichiarazione di tipo simile a typedef.

```
type MyContainer::const_iterator
```

Dichiarazione di un alias di tipo con tipo non specificato.

```
using MyType = std::unordered_map<int, std::string>
```

Dichiarazione di un alias di tipo.

```
template<typename T>  
using MyContainer = std::vector<T>
```

- .. **cpp:enum**:: unscoped enum declaration

.. `cpp:enum-struct::` scoped enum declaration

.. `cpp:enum-class::` scoped enum declaration

Describe un'enumerazione (con scope), possibilmente con il tipo specificato. Tutti gli enumeratori dichiarati all'interno di un'enumerazione senza scope verranno dichiarati sia nello scope dell'enumerazione che in quello genitore. Esempi:

```
.. cpp:enum:: MyEnum

    An unscoped enum.

.. cpp:enum:: MySpecificEnum : long

    An unscoped enum with specified underlying type.

.. cpp:enum-class:: MyScopedEnum

    A scoped enum.

.. cpp:enum-struct:: protected MyScopedVisibilityEnum : public
    std::underlying_type<MySpecificEnum>::type

    A scoped enum with non-default visibility, and with a specified
    underlying type.
```

.. `cpp:enumerator::` name

.. `cpp:enumerator::` name = constant

Describe un enumeratore, facoltativamente con il suo valore definito, ad esempio:

```
.. cpp:enumerator:: MyEnum::myEnumerator

.. cpp:enumerator:: MyEnum::myOtherEnumerator = 42
```

.. `cpp:union::` name

Describe una union.

Added in version 1.8.

.. `cpp:concept::` template-parameter-list name

Avvertimento: Il supporto per i concetti è sperimentale. Si basa sull'attuale bozza dello standard e sulle Concepts Technical Specification. Le funzionalità possono cambiare man mano che si evolvono.

Describe un concetto. Deve contenere esattamente 1 lista di parametri template. Il nome può essere un nome nidificato. Esempio:

```
.. cpp:concept:: template<typename It> std::Iterator
```

Proxy to an element of a notional sequence that can be compared, indirected, or incremented.

****Notation****

```
.. cpp:var:: It r
```

An lvalue.

****Valid Expressions****

- `:cpp:expr:`*r``, when `:cpp:expr:`r`` is dereferenceable.
- `:cpp:expr:`++r``, with return type `:cpp:expr:`It&``, when `:cpp:expr:`r`` is incrementable.

Ciò verrà visualizzato come segue:

```
template<typename It>
concept std::Iterator
```

Proxy di un elemento di una sequenza fittizia che può essere confrontata, indiretta o incrementata.

Notazione

It **r**

Un lvalue.

Espressioni Valide

- `*r`, quando `r` è de-referenziabile.
- `++r`, restituendo il tipo `It&`, quando `r` è incrementabile.

Added in version 1.5.

Opzioni

Alcune direttive hanno delle opzioni:

- `:no-index-entry:` e `:no-contents-entry:`, vedere *Markup di base*.
- `:tparam-line-spec:`, per le dichiarazioni con template. Se specificato, ogni parametro del template verrà visualizzato su una riga separata.

Added in version 1.6.

Entità Anonime

C++ supporta spazi dei nomi, classi, enumerazioni e unioni anonime. Ai fini della documentazione è necessario assegnare loro un nome che inizi con @, ad esempio, @42 o @data. Questi nomi possono essere utilizzati anche nei riferimenti incrociati e nelle espressioni (di tipo), sebbene i simboli annidati verranno trovati anche se omissi. Il nome @. . . apparirà sempre come **[anonymous]** (possibilmente come link).

Esempio:

```
.. cpp:class:: Data
    .. cpp:union:: @data
        .. cpp:var:: int a
        .. cpp:var:: double b

Explicit ref: cpp:var:`Data::@data::a`. Short-hand ref: cpp:var:`Data::a`.
↪cpp:var:`Data::a`.
```

Questo apparirà come:

```
class Data
    union [anonymous]
        int a
        double b
```

Esplicito ref: **Data::[anonymous]::a**. Abbreviato ref: **Data::a**.

Added in version 1.8.

Dichiarazioni di Alias

A volte può essere utile elencare le dichiarazioni altrove rispetto alla documentazione principale, ad esempio quando si crea una sinossi dell'interfaccia di una classe. A questo scopo può essere utilizzata la seguente direttiva.

```
.. cpp:alias:: name or function signature
```

Inserisce una o più dichiarazioni di alias. Ogni entità può essere specificata come nel ruolo **cpp:any**. Se viene fornito il nome di una funzione (invece della firma completa), verranno elencati tutti gli [overload] della funzione.

Per esempio:

```
.. cpp:alias:: Data::a
    overload_example::C::f
```

diventa

int *a*

void *f*(double d) const

void *f*(double d)

void *f*(int i)

void *f*()

mentre:

```
.. cpp:alias:: void overload_example::C::f(double d) const
           void overload_example::C::f(double d)
```

diventa

void *f*(double d) const

void *f*(double d)

Added in version 2.0.

Opzioni

:maxdepth: int

Si inseriscono anche dichiarazioni nidificate, fino alla profondità totale data. Utilizzare 0 per profondità infinita e 1 solo per la dichiarazione menzionata. Il default è 1.

Added in version 3.5.

:noroot:

Salta le dichiarazioni menzionate e visualizza solo le quelle annidate. Richiede `maxdepth` 0 o almeno 2.

Added in version 3.5.

Template Vincolati [Constrained]

Avvertimento: Il supporto per i concetti è sperimentale. Si basa sull'attuale bozza dello standard e sulle Concepts Technical Specification. Le funzionalità possono cambiare man mano che si evolvono.

Nota: Sphinx attualmente non supporta le clausole `requires`.

Segnaposti

Le dichiarazioni possono utilizzare il nome di un concetto per introdurre parametri template vincolati o la parola chiave `auto` per introdurre parametri template non vincolati:

```
.. cpp:function:: void f(auto &&arg)

    A function template with a single unconstrained template parameter.

.. cpp:function:: void f(std::Iterator it)

    A function template with a single template parameter, constrained_
    ↪by the
    Iterator concept.
```

Introduzioni dei Template

Semplici funzioni vincolate o classi template si possono dichiarare con una *template introduction* invece che con un elenco di parametri template:

```
.. cpp:function:: std::Iterator{It} void advance(It &it)

    A function template with a template parameter constrained to be an
    Iterator.

.. cpp:class:: std::LessThanComparable{T} MySortedContainer

    A class template with a template parameter constrained to be
    LessThanComparable.
```

Vengono resi come segue.

```
std::Iterator{It}
```

```
void advance(It &it)
```

Una funzione template con un parametro template vincolato ad essere un `Iterator`.

```
std::LessThanComparable{T}
```

```
class MySortedContainer
```

Una classe template con un parametro template vincolato ad essere `LessThanComparable`.

Si noti tuttavia che non viene eseguito alcun controllo rispetto alla compatibilità dei parametri. Ad esempio, `Iterator{A, B, C}` sarà accettato come introduzione anche se non sarebbe valido in C++.

Espressioni e Tipi In Linea

:cpp:expr:

:cpp:texpr:

Inserisce un'espressione C++ o un tipo sia come codice inline (`:cpp:expr`) che come testo inline (`:cpp:texpr`). Per esempio:

```
.. cpp:var:: int a = 42
```

```
.. cpp:function:: int f(int i)
```

An expression: `:cpp:expr:`a * f(a)`` (or as text: `:cpp:texpr:`a * f(a)``).

A type: `:cpp:expr:`const MySortedContainer<int>&``
(or as text `:cpp:texpr:`const MySortedContainer<int>&``).

apparirà in questo modo:

```
int a = 42
```

```
int f(int i)
```

Un'espressione: `a * f(a)` (o un testo: `a * f(a)`).

Un tipo: `const MySortedContainer<int>&` (o come testo `const MySortedContainer<int>&`).

Added in version 1.7: Il ruolo `cpp:expr`.

Added in version 1.8: Il ruolo `cpp:texpr`.

Lo Spazio dei Nomi

Per default le dichiarazioni nel dominio C++ vengono poste nello scope globale. Lo scope corrente può essere modificato utilizzando tre direttive sullo spazio dei nomi. Gestiscono uno stack di dichiarazioni dove `cpp:namespace` resetta lo stack e modifica un dato scope.

La direttiva `cpp:namespace-push` cambia lo scope in un dato scope interno a quello corrente.

La direttiva `cpp:namespace-pop` annulla la direttiva `cpp:namespace-push` più recente.

.. **cpp:namespace::** scope specification

Modifica lo scope corrente per gli oggetti successivi nell'ambito specificato e reimposta lo stack delle direttive dello spazio dei nomi. Da notare che lo spazio dei nomi non deve necessariamente corrispondere a quelli del C++, ma può terminare con nomi di classi, ad esempio:

```
.. cpp:namespace:: Namespace1::Namespace2::SomeClass::AnInnerClass
```


Tutti gli oggetti successivi verranno definiti come se il loro nome fosse dichiarato antepo-
nendo lo scope. I successivi riferimenti incrociati verranno ricercati a partire dallo scope
attuale.

Utilizzando `NULL`, `0` o `nullptr` lo scope cambierà nello Using `NULL`, `0` o `nullptr` come
scope modificherà quello globale.

Anche una dichiarazione di namespace può essere un template, ad esempio:

```
.. cpp:class:: template<typename T> \  
    std::vector  
  
.. cpp:namespace:: template<typename T> std::vector  
  
.. cpp:function:: std::size_t size() const
```

dichiara `size` come funzione membro della classe template `std::vector`. In modo
equivalente si poteva dichiarare con:

```
.. cpp:class:: template<typename T> \  
    std::vector  
  
.. cpp:function:: std::size_t size() const
```

oppure:

```
.. cpp:class:: template<typename T> \  
    std::vector
```

.. **cpp:namespace-push::** scope specification

Modifica lo scope relativamente a quello corrente. Per esempio, dopo:

```
.. cpp:namespace:: A::B  
  
.. cpp:namespace-push:: C::D
```

lo scope corrente sarà `A::B::C::D`.

Added in version 1.4.

.. **cpp:namespace-pop::**

Annulla la precedente direttiva `cpp:namespace-push` (*non* il semplice 'pop' di uno
scope). Per esempio, dopo:

```
.. cpp:namespace:: A::B  
  
.. cpp:namespace-push:: C::D  
  
.. cpp:namespace-pop::
```

lo scope corrente sarà `A::B` (*not* `A::B::C`).

Se non è stata utilizzata alcuna direttiva `cpp:namespace-push` precedente, ma solo una direttiva `cpp:namespace`, lo scope corrente verrà reimpostato su quello globale. Cioè, `.. cpp:namespace:: A::B` è equivalente a:

```
.. cpp:namespace:: nullptr
.. cpp:namespace-push:: A::B
```

Added in version 1.4.

Liste di campi info

Tutte le direttive C++ per la dichiarazione delle entità supportano i seguenti campi info (consultare anche *Liste di campi info*):

- `tparam`: Descrizione di un parametro template.

La direttiva `cpp:function` supporta inoltre i seguenti campi:

- `param`, `parameter`, `arg`, `argument`: Descrizione di un parametro.
- `returns`, `return`: Descrizione di un valore restituito.
- `retval`, `retvals`: Un'alternativa a `returns` per descrivere il risultato della funzione.
- `throws`, `throw`, `exception`: Descrizione di un possibile errore generato.

Added in version 4.3: Il tipo del campo `retval`.

Riferimenti incrociati

Questi ruoli si collegano ai tipi di dichiarazione specificati:

`:cpp:any:`
`:cpp:class:`
`:cpp:struct:`
`:cpp:func:`
`:cpp:member:`
`:cpp:var:`
`:cpp:type:`
`:cpp:concept:`
`:cpp:enum:`
`:cpp:enumerator:`

Riferimento a una dichiarazione C++ per nome (vedere di seguito per i dettagli). Il nome deve essere adeguatamente qualificato rispetto alla posizione del link.

Added in version 2.0: Il ruolo `cpp:struct` come alias per il ruolo `cpp:class`.

Nota sui Riferimenti con Parametri/Argomenti Template

Questi ruoli seguono le regole Sphinx *Sintassi dei riferimenti incrociati*. Ciò significa che è necessario prestare attenzione quando si fa riferimento a una specializzazione (parziale) del template, ad es. se il link assomiglia a questo: `:cpp:class:`MyClass<int>``. Questo viene interpretato come un collegamento a `int` col titolo `MyClass`. In questo caso, si esegue l'escape della parentesi angolari di apertura con una barra rovesciata, in questo modo: `:cpp:class:`MyClass\<int>``.

Quando non è necessario un titolo personalizzato, potrebbero risultare utili i ruoli per le espressioni inline, `cpp:expr` e `cpp:texpr`, dove le parentesi angolari non necessitano di escape.

Dichiarazioni senza parametri template e argomenti template

Per il collegamento a dichiarazioni non basate su template, il nome deve essere un nome annidato, ad esempio, `f` o `MyClass::f`.

Funzioni (membro) sovraccaricate [overloaded]

Quando si fa riferimento a una funzione (membro) utilizzando solo il nome, il riferimento punterà a un overload di corrispondenza arbitraria. I ruoli `cpp:any` e `cpp:func` utilizzano un formato alternativo, che è semplicemente una dichiarazione di funzione completa. Ciò si risolverà nell'overlay corrispondente esatto. Ad esempio, consideriamo la seguente dichiarazione di classe:

```
class C
```

```
    void f(double d) const
```

```
    void f(double d)
```

```
    void f(int i)
```

```
    void f()
```

Riferimento col ruolo `cpp:func`:

- Overload arbitrario: `C::f`, `C::f()`
- Anche overload arbitrario: `C::f()`, `C::f()`
- Overload specifico: `void C::f()`, `void C::f()`
- Overload specifico: `void C::f(int)`, `void C::f(int)`
- Overload specifico: `void C::f(double)`, `void C::f(double)`
- Overload specifico: `void C::f(double) const`, `void C::f(double) const`

Da notare che la variabile di configurazione `add_function_parentheses` non influenza i riferimenti degli overload specifici.

Dichiarazioni basate sui Template

Si suppongano le seguenti dichiarazioni.

```
class Wrapper
```

```
    template<typename TOuter>  
    class Outer
```

```
        template<typename TInner>  
        class Inner
```

In generale il riferimento deve includere le dichiarazioni dei parametri template e gli argomenti template per il prefisso dei nomi qualificati. Per esempio:

- `template<typename TOuter> Wrapper::Outer (template<typename TOuter> Wrapper::Outer)`
- `template<typename TOuter> template<typename TInner> Wrapper::Outer<TOuter>::Inner (template<typename TOuter> template<typename TInner> Wrapper::Outer<TOuter>::Inner)`

Attualmente la ricerca ha esito positivo solo se gli identificatori dei parametri template sono stringhe uguali. Cioè, `template<typename UOuter> Wrapper::Outer` non funzionerà.

In breve, se l'elenco dei parametri template viene omissso, la ricerca presupporrà un template primario o un non template, ma non una specializzazione parziale del template. Ciò significa che funzionano anche i seguenti riferimenti:

- `Wrapper::Outer (Wrapper::Outer)`
- `Wrapper::Outer::Inner (Wrapper::Outer::Inner)`
- `template<typename TInner> Wrapper::Outer::Inner (template<typename TInner> Wrapper::Outer::Inner)`

Specializzazioni Template (Completo)

Si suppongano le seguenti dichiarazioni.

```
template<typename TOuter>  
class Outer
```

```
    template<typename TInner>  
    class Inner
```

```
template<>
```

```
class Outer<int>

    template<typename TInner>
    class Inner

    template<>
    class Inner<bool>
```

In generale il riferimento deve includere un elenco di parametri template per ogni elenco di argomenti template. È quindi possibile fare riferimento alla specializzazione completa di cui sopra con `template\<> Outer\<int>` (`template\<> Outer\<int>`) e `template\<> template\<> Outer\<int>::Inner\<bool>` (`template\<> template\<> Outer\<int>::Inner\<bool>`). In breve, l'elenco vuoto dei parametri template può essere omesso, ad esempio, `Outer\<int>` (`Outer\<int>`) and `Outer\<int>::Inner\<bool>` (`Outer\<int>::Inner\<bool>`).

Specializzazioni Template Parziali

Si suppongano le seguenti dichiarazioni.

```
template<typename T>
class Outer<T*>
```

I riferimenti a specializzazioni parziali devono sempre includere gli elenchi di parametri template, ad esempio, `template\<typename T> Outer\<T*>` (`template\<typename T> Outer\<T*>`). Attualmente la ricerca ha esito positivo solo se gli identificatori dei parametri template sono stringhe uguali.

Variabili di Configurazione

Vedere *Opzioni per il dominio C++*.

Il Dominio JavaScript

Added in version 1.0.

Il dominio JavaScript (nome **js**) fornisce le seguenti direttive:

.. js:module:: name

Questa direttiva imposta il nome del modulo per le dichiarazioni degli oggetti che seguono. Il nome del modulo viene utilizzato nell'indice globale dei moduli e nei riferimenti incrociati. Questa direttiva non crea un'intestazione di un oggetto come farebbe, ad esempio *py:class*.

Per default, questa direttiva creerà un'entità linkabile e inserirà una voce nell'indice globale dei moduli, a meno che non venga specificata l'opzione `no-index`. Se questa opzione è specificata, la direttiva aggiornerà solo il nome del modulo corrente.

Added in version 1.6.

Cambiato nella versione 5.2: Le direttive del modulo a supporto del contenuto del corpo.

.. js:function:: name(signature)

Descrive una funzione o un metodo JavaScript. Per descrivere gli argomenti come facoltativi si usano le parentesi quadre come *documented* per le signature Python.

Si possono utilizzare i campi per fornire maggiori dettagli sugli argomenti e sui tipi previsti, sugli errori che possono essere generati dalla funzione e sul valore restituito:

```
.. js:function:: $.getJSON(href, callback[, errback])

:param string href: An URI to the location of the resource.
:param callback: Gets called with the object.
:param errback:
    Gets called in case the request fails. And a lot of other
    text so we need multiple lines.
:throws SomeError: For whatever reason in that case.
:returns: Something.
```

Questo apparirà come:

```
$.getJSON(href, callback[, errback ])
```

Parametri

- **href** (string()) -- Un URI per la posizione della risorsa.
- **callback** -- Viene chiamato con l'oggetto.
- **errback** -- Viene chiamata nel caso in cui la richiesta fallisca. E molto altro testo, quindi abbiamo bisogno di più righe.

Solleva

SomeError() -- Per qualunque motivo in quel caso.

Ritorna

Something [*qualcosa*].

:single-line-parameter-list: (no value)

Garantisce che i parametri della funzione verranno emessi su una singola riga logica, sovrascrivendo (overriding) *javascript_maximum_signature_line_length* e *maximum_signature_line_length*.

Added in version 7.1.

.. js:method:: name(signature)

Questa direttiva è un alias per *js:function*, tuttavia descrive una funzione implementata come metodo su un oggetto di una classe.

Added in version 1.6.

:single-line-parameter-list: (no value)

Garantisce che i parametri della funzione verranno emessi su una singola riga logica,

sovrascrivendo (overriding) `javascript_maximum_signature_line_length` e `maximum_signature_line_length`.

Added in version 7.1.

.. js:class:: name

Descrive un costruttore che crea un oggetto. Fondamentalmente è come una funzione ma verrà visualizzata con il prefisso *class*:

```
.. js:class:: MyAnimal(name[, age])

:param string name: The name of the animal
:param number age: an optional age for the animal
```

Questo apparirà come:

```
class MyAnimal(name[, age])
```

Parametri

- **name** (string()) -- Il nome dell'animale
- **age** (number()) -- un'età facoltativa per l'animale

:single-line-parameter-list: (no value)

Garantisce che i parametri della funzione verranno emessi su una singola riga logica, sovrascrivendo (overriding) `javascript_maximum_signature_line_length` e `maximum_signature_line_length`.

Added in version 7.1.

.. js:data:: name

Descrive una variabile o costante globale.

.. js:attribute:: object.name

Descrive l'attributo *name* di *object*.

Questi ruoli sono forniti per fare riferimento agli oggetti descritti:

```
:js:mod:
:js:func:
:js:meth:
:js:class:
:js:data:
:js:attr:
```

Il dominio della Matematica

Added in version 1.8.

Il dominio matematico (nome **math**) prevede i seguenti ruoli:

:math:numref:

Ruolo per i riferimenti incrociati delle equazioni definiti dalla direttiva *math* tramite la loro etichetta. Esempio:

```
.. math:: e^{\mathrm{i}\pi} + 1 = 0
   :label: euler
```

Euler's identity, equation `:math:numref:`euler``, was elected one of the most beautiful mathematical formulas.

Added in version 1.8.

Il Dominio Python

Added in version 1.0.

Il dominio Python (nome **py**) fornisce le seguenti direttive per le dichiarazioni dei moduli:

.. py:module:: name

Questa direttiva segna l'inizio della descrizione di un modulo (o del sottomodulo del pacchetto, nel qual caso il nome dovrebbe essere completo, incluso il nome del pacchetto). Una descrizione del modulo come la docstring può essere inserita nel corpo della direttiva.

Questa direttiva causerà anche una voce nell'indice globale del modulo.

Cambiato nella versione 5.2: Le direttive del modulo a supporto del contenuto del corpo.

opzioni

:platform: platforms (comma separated list)

Indica le piattaforme su cui è disponibile il modulo (se è disponibile su tutte le piattaforme, l'opzione deve essere omessa). Le chiavi sono identificatori brevi; esempi in uso includono "IRIX", "Mac", "Windows" e "Unix". È importante utilizzare una chiave già utilizzata, quando applicabile.

:synopsis: purpose (text)

Consiste in una frase che descrive lo scopo del modulo: attualmente è utilizzato solo nell'indice globale dei moduli.

:deprecated: (no argument)

Contrassegna un modulo come deprecato; allora sarà designato come tale in varie località.

.. py:currentmodule:: name

Questa direttiva dice a Sphinx che le classi, le funzioni ecc. documentate da qui si trovano nel modulo specificato (come [py:module](#)), ma non creerà voci di indice, una voce nell'indice del modulo globale o un link al target per [py:mod](#). Questo è utile in situazioni in cui la documentazione di elementi in un modulo è distribuita su più file o sezioni: una posizione ha la direttiva [py:module](#), le altre solo [py:currentmodule](#).

Per i contenuti dei moduli e delle classi vengono fornite le seguenti direttive:

.. py:function:: name(parameters)

.. py:function:: name[type parameters](parameters)

Descrive una funzione a livello di modulo. La firma dovrebbe includere i parametri, insieme ai parametri di tipo opzionali, come fornito nella definizione della funzione Python, vedere *Le Signature Python*. Per esempio:

```
.. py:function:: Timer.repeat(repeat=3, number=1_000_000)
.. py:function:: add[T](a: T, b: T) -> T
```

Per i metodi si deve usare [py:method](#).

La descrizione normalmente include informazioni sui parametri richiesti e su come vengono utilizzati (in particolare se gli oggetti mutabili passati come parametri vengono modificati), effetti collaterali e possibili eccezioni.

Questa informazione può essere (in qualsiasi direttiva py) opzionalmente fornita in una forma strutturata, vedere *Liste di campi info*.

opzioni

:async: (no value)

Indica che la funzione è una funzione asincrona.

Added in version 2.1.

:canonical: (full qualified name including module name)

Descrivere la posizione in cui è definito l'oggetto se è importato da altri moduli

Added in version 4.0.

:single-line-parameter-list: (no value)

Garantisce che gli argomenti della funzione verranno emessi su una singola riga logica, sovrascrivendo [python_maximum_signature_line_length](#) e [maximum_signature_line_length](#).

Added in version 7.1.

:single-line-type-parameter-list: (no value)

Assicura che i parametri del tipo della funzione vengano emessi su una singola linea logica, sovrascrivendo [python_maximum_signature_line_length](#) e [maximum_signature_line_length](#).

Added in version 7.1.

.. py:data:: name

Descrive i dati globali in un modulo, incluse sia le variabili che i valori utilizzati come le "costanti definite". Gli attributi di classe e oggetto non sono documentati utilizzando questo ambiente.

opzioni

:type: type of the variable (text)

Added in version 2.4.

:value: initial value of the variable (text)

Added in version 2.4.

:canonical: (full qualified name including module name)

Descrivere la posizione in cui è definito l'oggetto se è importato da altri moduli

Added in version 4.0.

.. py:exception:: name

.. py:exception:: name(parameters)

.. py:exception:: name[type parameters](parameters)

Descrive una classe di eccezione. La signature [*firma*] può, ma non necessariamente, includere parentesi con gli argomenti del costruttore o può facoltativamente includere parametri di tipo (vedere [PEP 695](https://peps.python.org/pep-0695/)²⁶⁴).

opzioni

:final: (no value)

Indica che è una classe è 'final'.

Added in version 3.1.

:single-line-parameter-list: (no value)

See [py:class:single-line-parameter-list](#).

Added in version 7.1.

:single-line-type-parameter-list: (no value)

See [py:class:single-line-type-parameter-list](#).

Added in version 7.1.

.. py:class:: name

.. py:class:: name(parameters)

²⁶⁴ <https://peps.python.org/pep-0695/>

.. py:class:: name[type parameters](parameters)

Descrive una classe. La firma può opzionalmente includere parametri di tipo (vedere [PEP 695](#)²⁶⁵) o parentesi con parametri che verranno mostrati come argomenti del costruttore. Vedere anche *Le Signature Python*.

I metodi e gli attributi appartenenti alla classe dovrebbero essere inseriti nel corpo di questa direttiva. Se vengono posizionati all'esterno, il nome fornito dovrebbe contenere il nome della classe in modo che i riferimenti incrociati funzionino ancora. Esempio:

```
.. py:class:: Foo

    .. py:method:: quux()

-- or --

.. py:class:: Bar

.. py:method:: Bar.quux()
```

Il primo metodo è quello preferito.

opzioni

:canonical: (full qualified name including module name)

Descrivere la posizione in cui è definito l'oggetto se è importato da altri moduli

Added in version 4.0.

:final: (no value)

Indica che è una classe è 'final'.

Added in version 3.1.

:single-line-parameter-list: (no value)

Garantisce che gli argomenti del costruttore della classe verranno emessi su una singola riga logica, sovrascrivendo *python_maximum_signature_line_length* e *maximum_signature_line_length*.

Added in version 7.1.

:single-line-type-parameter-list: (no value)

Assicura che i parametri del tipo di classe vengano emessi su una singola linea logica, sovrascrivendo *python_maximum_signature_line_length* e *maximum_signature_line_length*.

.. py:attribute:: name

Descrive un attributo dei dati dell'oggetto. La descrizione dovrebbe includere informazioni sul tipo di dati previsti e se possono essere modificati direttamente.

²⁶⁵ <https://peps.python.org/pep-0695/>

opzioni

:type: type of the attribute (text)

Added in version 2.4.

:value: initial value of the attribute (text)

Added in version 2.4.

:canonical: (full qualified name including module name)

Descrivere la posizione in cui è definito l'oggetto se è importato da altri moduli

Added in version 4.0.

.. py:property:: name

Descrive la proprietà di un oggetto.

Added in version 4.0.

opzioni

:abstractmethod: (no value)

Indica che la proprietà è astratta.

:classmethod: (no value)

Indica che la proprietà è un metodo di classe.

Added in version 4.2.

:type: type of the property (text)

.. py:method:: name(parameters)

.. py:method:: name[type parameters](parameters)

Descrive un metodo dell'oggetto. I parametri non devono includere il parametro `self`. La descrizione dovrebbe includere informazioni simili a quelle descritte per la `function`. Vedere anche *Le Signature Python* e *Liste di campi info*.

opzioni

:abstractmethod: (no value)

Indica che il metodo è astratto.

Added in version 2.1.

:async: (no value)

Indica che il metodo è asincrono.

Added in version 2.1.

:canonical: (full qualified name including module name)

Descrivere la posizione in cui è definito l'oggetto se è importato da altri moduli

Added in version 4.0.

:classmethod: (no value)

Indicare che il metodo è un metodo di classe.

Added in version 2.1.

:final: (no value)

Indica che il metodo è un metodo finale.

Added in version 3.1.

:single-line-parameter-list: (no value)

Garantisce che gli argomenti del metodo verranno emessi su una singola riga logica, sovrascrivendo `python_maximum_signature_line_length` e `maximum_signature_line_length`.

Added in version 7.1.

:single-line-type-parameter-list: (no value)

Assicura che i parametri del tipo del metodo vengano emessi su una singola linea logica, sovrascrivendo `python_maximum_signature_line_length` e `maximum_signature_line_length`.

Added in version 7.2.

:staticmethod: (no value)

Indica che il metodo è statico.

Added in version 2.1.

.. py:staticmethod:: name(parameters)

.. py:staticmethod:: name[type parameters](parameters)

Come `py:method`, ma indica che il metodo è statico.

Added in version 0.4.

.. py:classmethod:: name(parameters)

.. py:classmethod:: name[type parameters](parameters)

Come `py:method`, ma indica che il metodo è un metodo di classe.

Added in version 0.6.

.. py:decorator:: name

.. py:decorator:: name(parameters)

.. py:decorator:: name[type parameters](parameters)

Descrive una funzione di decorazione [decorator]. La signature (*firma*) dovrebbe rappresentare l'utilizzo come decoratore. Ad esempio, date le funzioni

```
def removename(func):
    func.__name__ = ''
    return func

def setnewname(name):
    def decorator(func):
        func.__name__ = name
        return func
    return decorator
```

le descrizioni dovrebbero assomigliare a queste:

```
.. py:decorator:: removename

    Remove name of the decorated function.

.. py:decorator:: setnewname(name)

    Set name of the decorated function to *name*.
```

(al contrario di `.. py:decorator:: removename(func)`).

Non esiste un ruolo `py:deco` da collegare a un decoratore contrassegnato con questa direttiva; piuttosto, si usa il ruolo `py:func`.

:single-line-parameter-list: (no value)

Garantisce che gli argomenti del decoratore verranno emessi su una singola riga logica, sovrascrivendo `python_maximum_signature_line_length` e `maximum_signature_line_length`.

Added in version 7.1.

:single-line-type-parameter-list: (no value)

Assicura che i parametri di tipo del decoratore vengano emessi su una singola riga logica, sovrascrivendo `python_maximum_signature_line_length` e `maximum_signature_line_length`.

Added in version 7.2.

```
.. py:decoratormethod:: name
.. py:decoratormethod:: name(signature)
.. py:decoratormethod:: name[type parameters](signature)
```

Lo stesso di `py:decorator`, ma per i decoratori che sono metodi.

Riferimento a un metodo decoratore utilizzando il ruolo `py:meth`.

Le Signature Python

Le signature [*firme*] di funzioni, metodi e costruttori di classi possono essere fornite come se fossero scritte in Python.

Si possono fornire i valori di default per gli argomenti opzionali (ma se contengono virgole, confonderanno il parser). Possono essere fornite anche annotazioni sugli argomenti in stile Python 3 così come le annotazioni sul tipo restituito:

```
.. py:function:: compile(source : string, filename, symbol='file') ->
   ↳ ast object
```

Per le funzioni con parametri facoltativi che non hanno valori di default (tipicamente funzioni implementate nei moduli di estensione C senza supporto di argomenti con parole chiave), si possono utilizzare le parentesi per specificare le parti facoltative:

```
compile(source[, filename[, symbol]])
```

È consuetudine aprire la parentesi prima della virgola.

Python 3.12 ha introdotto i *parametri di tipo*, che sono variabili di tipo dichiarate direttamente all'interno della definizione di classe o funzione:

```
class AnimalList[AnimalT](list[AnimalT]):
    ...

def add[T](a: T, b: T) -> T:
    return a + b
```

La documentazione reStructuredText corrispondente sarebbe:

```
.. py:class:: AnimalList[AnimalT]

.. py:function:: add[T](a: T, b: T) -> T
```

Vedere [PEP 695](https://peps.python.org/pep-0695/)²⁶⁶ e [PEP 696](https://peps.python.org/pep-0696/)²⁶⁷ per i dettagli e le specifiche complete.

Liste di campi info

Added in version 0.4.

Cambiato nella versione 3.0: aggiunti i meta campi.

Nelle direttive di descrizione degli oggetti Python, gli elenchi di campi reST con questi campi vengono riconosciuti e formattati correttamente:

- `param`, `parameter`, `arg`, `argument`, `key`, `keyword`: Descrizione di un parametro.
- `type`: Tipo di un parametro. Crea un link se possibile.

²⁶⁶ <https://peps.python.org/pep-0695/>

²⁶⁷ <https://peps.python.org/pep-0696/>

- `raises, raise, except, exception`: Viene sollevata un'eccezione (e quando).
- `var, ivar, cvar`: Descrizione di una variabile.
- `vartype`: Tipo di una variabile. Crea un link se possibile.
- `returns, return`: Descrizione del valore restituito.
- `rtype`: Tipo restituito. Crea un link se possibile.
- `meta`: Aggiunge metadati alla descrizione dell'oggetto Python. I metadati non verranno visualizzati sul documento di output. Per esempio, `:meta private:` indica che l'oggetto Python è un membro privato. Viene utilizzato in [sphinx.ext.autodoc](#) per filtrare i membri.

Nota: Nella versione attuale, tutti i `var`, `ivar` e `cvar` sono rappresentati come "Variable". Non c'è alcuna differenza.

I nomi dei campi devono essere costituiti da una di queste parole chiave e da un argomento (ad eccezione di `returns` e `rtype`, che non necessitano di argomenti). Questi è spiegato meglio con un esempio:

```
.. py:function:: send_message(sender, recipient, message_body,
    ↳[priority=1])

    Send a message to a recipient

    :param str sender: The person sending the message
    :param str recipient: The recipient of the message
    :param str message_body: The body of the message
    :param priority: The priority of the message, can be a number 1-5
    :type priority: integer or None
    :return: the message id
    :rtype: int
    :raises ValueError: if the message_body exceeds 160 characters
    :raises TypeError: if the message_body is not a basestring
```

Questo verrà visualizzato in questo modo:

send_message(*sender*, *recipient*, *message_body*[, *priority=1*])

Invia un messaggio a un destinatario

Parametri

- **sender** (*str*²⁶⁸) -- La persona che invia il messaggio
- **recipient** (*str*²⁶⁹) -- Il destinatario del messaggio
- **message_body** (*str*²⁷⁰) -- Il corpo del messaggio
- **priority** (*int*²⁷¹ or *None*) -- La priorità del messaggio può essere un numero compreso tra 1 e 5

Ritorna

l'ID del messaggio

Tipo di ritorno`int`²⁷²**Solleva**

- **ValueError**²⁷³ -- se il `message_body` supera i 160 caratteri
- **TypeError**²⁷⁴ -- se il `message_body` non è una `basestring`

È anche possibile combinare tipo e descrizione del parametro, se il tipo è una sola parola, in questo modo:

```
:param int priority: The priority of the message, can be a number 1-5
```

Added in version 1.5.

I tipi di contenitori come le liste e i dizionari possono essere collegati automaticamente utilizzando la seguente sintassi:

```
:type priorities: list(int)
:type priorities: list[int]
:type mapping: dict(str, int)
:type mapping: dict[str, int]
:type point: tuple(float, float)
:type point: tuple[float, float]
```

Tipi multipli in un campo di tipo verranno collegati automaticamente se separati dalla parola "or":

```
:type an_arg: int or None
:var type a_var: str or int
:rtype: float or str
```

²⁶⁸ <https://docs.python.org/3/library/stdtypes.html#str>

²⁶⁹ <https://docs.python.org/3/library/stdtypes.html#str>

²⁷⁰ <https://docs.python.org/3/library/stdtypes.html#str>

²⁷¹ <https://docs.python.org/3/library/functions.html#int>

²⁷² <https://docs.python.org/3/library/functions.html#int>

²⁷³ <https://docs.python.org/3/library/exceptions.html#ValueError>

²⁷⁴ <https://docs.python.org/3/library/exceptions.html#TypeError>

Riferimenti incrociati di oggetti Python

I seguenti ruoli si riferiscono agli oggetti nei moduli e sono eventualmente collegati tramite link ipertestuale se viene trovato un identificatore corrispondente:

:py:mod:

Riferimento a un modulo; è possibile utilizzare un nome puntato [dotted]. Questo dovrebbe essere usato anche per i nomi dei pacchetti.

:py:func:

Riferimento a una funzione Python; possono essere usati nomi puntati [dotted]. Non è necessario che il testo del ruolo includa parentesi finali per migliorare la leggibilità; verranno aggiunti automaticamente da Sphinx se il valore di configurazione [add_function_parentheses](#) è True (il default).

:py:data:

Riferimento a una variabile a livello di modulo.

:py:const:

Riferimento a una costante "definita". Potrebbe trattarsi di una variabile Python che non è destinata a essere modificata.

:py:class:

Riferimento a una classe; è possibile utilizzare un nome puntato [dotted].

:py:meth:

Riferimento a un metodo di un oggetto. Il testo del ruolo può includere il nome del tipo e quello del metodo; se ricorre nella descrizione di un tipo, il nome del tipo può essere omissso. Si può usare un nome 'dotted'.

:py:attr:

Riferimento a un attributo dati di un oggetto.

Nota: Il ruolo può anche riferirsi alla proprietà.

:py:exc:

Riferimento a un'eccezione. Si può usare un nome 'dotted'.

:py:obj:

Riferimento a un oggetto di tipo non specificato. Utile ad es. come [default_role](#).

Added in version 0.4.

Il nome racchiuso in questo markup può includere il nome di un modulo e/o il nome di una classe. Ad esempio, `:py:func:`filter`` potrebbe fare riferimento a una funzione denominata `filter` nel modulo corrente o alla funzione nativa con quel nome. Al contrario, `:py:func:`foo.filter`` si riferisce chiaramente alla funzione `filter` nel modulo `foo`.

Normalmente, i nomi in questi ruoli vengono cercati prima senza alcuna ulteriore qualificazione, poi con il nome del modulo corrente anteposto, quindi con il nome del modulo corrente e della classe (se presente) anteposti. Se si aggiunge un punto al nome come prefisso

so, l'ordine viene invertito. Per esempio, nella documentazione del modulo `codecs`²⁷⁵ di Python, `:py:func:`open`` i riferisce sempre alla funzione nativa, mentre `:py:func:`.open`` si riferisce a `codecs.open()`²⁷⁶.

Un'euristica simile viene utilizzata per determinare se il nome è un attributo della classe attualmente documentata.

Inoltre, se il nome ha come prefisso un punto e non viene trovata alcuna corrispondenza esatta, la destinazione viene considerata come suffisso e vengono cercati tutti i nomi di oggetto con quel suffisso. Per esempio, `:py:meth:`.TarFile.close`` fa riferimento alla funzione `tarfile.TarFile.close()`, anche se il modulo corrente non è `tarfile`. Poiché questo può diventare ambiguo, se esiste più di una corrispondenza possibile, si riceverà un warning da Sphinx.

Notare che si possono combinare i prefissi `~` e `.`: `:py:meth:`~.TarFile.close`` farà riferimento al metodo `tarfile.TarFile.close()`, ma la didascalia visibile del link sarà solo `close()`.

Il Dominio reStructuredText

Added in version 1.0.

Il dominio reStructuredText (nome **rst**) fornisce le seguenti direttive:

.. rst:directive:: name

Descrive una direttiva reST. Il *nome* può essere il nome di una singola direttiva o la sintassi effettiva della direttiva (prefisso `..` e suffisso `::`) con argomenti che verranno resi in modo diverso. Per esempio:

```
.. rst:directive:: foo

    Foo description.

.. rst:directive:: .. bar:: baz

    Bar description.
```

apparirà come:

```
.. foo::
    Descrizione di foo.

.. bar:: baz
    Descrizione di bar.
```

.. rst:directive:option:: name

Descrive un'opzione per la direttiva reST. Il *nome* può essere un nome di un'opzione singola o un nome di opzione con argomenti separati da due punti (`:`). Per esempio:

²⁷⁵ <https://docs.python.org/3/library/codecs.html#module-codecs>

²⁷⁶ <https://docs.python.org/3/library/codecs.html#codecs.open>

```
.. rst:directive:: toctree

   .. rst:directive:option:: caption: caption of ToC

   .. rst:directive:option:: glob
```

apparirà come:

```
.. toctree::

   :caption:  caption of ToC

   :glob:
```

opzioni

:type: description of argument (text)

Descrive il tipo di valore dell'opzione.

Per esempio:

```
.. rst:directive:: toctree

   .. rst:directive:option:: maxdepth
      :type: integer or no value
```

Added in version 2.1.

.. rst:role:: name

Descrive un ruolo reST. Per esempio:

```
.. rst:role:: foo

   Foo description.
```

apparirà come:

```
:foo:

   Descrizione di foo.
```

Questi ruoli sono forniti per fare riferimento agli oggetti descritti:

:rst:dir:

:rst:role:

Altri domini

Sono disponibili diversi domini di terze parti come estensioni, tra cui:

- [Ada](#)²⁷⁷
- [Chapel](#)²⁷⁸
- [CoffeeScript](#)²⁷⁹
- [Common Lisp](#)²⁸⁰
- [dqn](#)²⁸¹
- [Erlang](#)²⁸²
- [Go](#)²⁸³
- [HTTP](#)²⁸⁴
- [Jinja](#)²⁸⁵
- [Lasso](#)²⁸⁶
- [MATLAB](#)²⁸⁷
- [Operation](#)²⁸⁸
- [PHP](#)²⁸⁹
- [Ruby](#)²⁹⁰
- [Scala](#)²⁹¹

²⁷⁷ <https://pypi.org/project/sphinxcontrib-adadomain/>

²⁷⁸ <https://pypi.org/project/sphinxcontrib-chapeldomain/>

²⁷⁹ <https://pypi.org/project/sphinxcontrib-coffee/>

²⁸⁰ <https://pypi.org/project/sphinxcontrib-cldomain/>

²⁸¹ <https://pypi.org/project/sphinxcontrib-dqndomain/>

²⁸² <https://pypi.org/project/sphinxcontrib-erlangdomain/>

²⁸³ <https://pypi.org/project/sphinxcontrib-golangdomain/>

²⁸⁴ <https://pypi.org/project/sphinxcontrib-httpdomain/>

²⁸⁵ <https://pypi.org/project/sphinxcontrib-jinjadomain/>

²⁸⁶ <https://pypi.org/project/sphinxcontrib-lassodomain/>

²⁸⁷ <https://pypi.org/project/sphinxcontrib-matlabdomain/>

²⁸⁸ <https://pypi.org/project/sphinxcontrib-operationdomain/>

²⁸⁹ <https://pypi.org/project/sphinxcontrib-phpdomain/>

²⁹⁰ <https://pypi.org/project/sphinxcontrib-rubydomain/>

²⁹¹ <https://pypi.org/project/sphinxcontrib-scaladomain/>

4.7 Estensioni

Poiché molti progetti avranno bisogno di funzionalità speciali nella loro documentazione, Sphinx consente di aggiungere "extensions" al processo di creazione, ognuna delle quali può modificare quasi ogni aspetto dell'elaborazione dei documenti.

Questo capitolo descrive le estensioni incluse in Sphinx. Per la documentazione API su come scrivere la propria estensione, fare riferimento a *Le API di Sphinx*.

Estensioni integrate

Queste estensioni sono native e possono essere attivate dalle rispettive voci nel valore di configurazione `extensions`:

`sphinx.ext.autodoc` -- Include la documentazione da docstring

Questa estensione può importare i moduli che si stanno documentando e inserire la documentazione da docstring in modo semiautomatico.

Avvertimento: `autodoc` importa i moduli da documentare. Se qualche modulo ha effetti collaterali durante l'importazione, questi verranno eseguiti da `autodoc` quando viene eseguito `sphinx-build`.

Se si documentano gli script (al contrario dei moduli della libreria), la loro routine principale dev'essere protetta da una condizione `if __name__ == '__main__':`.

Affinché ciò funzioni, le docstring devono ovviamente essere scritte nel correttamente in reStructuredText. Si può poi utilizzare tutto il solito markup Sphinx nelle docstring e finirà correttamente nella documentazione. Insieme alla documentazione scritta a mano, questa tecnica allevia il problema di dover mantenere due posizioni per la documentazione, evitando allo stesso tempo una documentazione API dall'aspetto 'meccanico'.

Se si preferiscono le docstring in stile NumPy²⁹² o Google²⁹³ rispetto a reStructuredText, si può anche abilitare l'estensione `napoleon`. `napoleon` è un preprocessore che converte le docstring in un reStructuredText corretto prima che `autodoc` le elabori.

²⁹² <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

²⁹³ <https://google.github.io/styleguide/pyguide.html#38-comments-and-docstrings>

Per iniziare

Setup

Attivare il plugin aggiungendo 'sphinx.ext.autodoc' nelle *extensions* in `conf.py`:

```
extensions = [  
    ...  
    'sphinx.ext.autodoc',  
]
```

Garantire che il codice possa essere importato

autodoc analizza il codice e le docstring mediante introspezione dopo aver importato i moduli. Affinché l'importazione funzioni, i moduli devono poter essere trovati da Sphinx e che le dipendenze possano essere risolte (se il modulo esegue `import foo`, ma `foo` non è disponibile nell'ambiente python in cui Sphinx viene eseguito, l'importazione del modulo fallirà).

Esistono due modi per garantire ciò:

1. Utilizzare un ambiente che contenga il pacchetto e Sphinx. Questo può ad es. essere l'ambiente di sviluppo locale (con un'installazione modificabile) o un ambiente in CI in cui si installa Sphinx e il pacchetto. Il regolare processo di installazione garantisce che il pacchetto possa essere trovato da Sphinx e che tutte le dipendenze siano disponibili.
2. In alternativa è possibile patchare il run di Sphinx in modo che possa operare direttamente sui sorgenti; per esempio. se si vuol essere in grado di eseguire una build Sphinx da un checkout del sorgente.
 - "Patchare" `sys.path`²⁹⁴ nel `conf.py` di Sphinx per includere la cartella dei sorgenti. Per esempio. se si ha una struttura di repository con `doc/conf.py` e il pacchetto si trova in `src/mypackage`, allora si deve aggiungere:

```
sys.path.insert(0, os.path.abspath('../src'))
```

a `conf.py`.

- Per far fronte alle dipendenze mancanti, specificare i moduli mancanti in *autodoc_mock_imports*.

²⁹⁴ <https://docs.python.org/3/library/sys.html#sys.path>

Utilizzo

Ora si può usare *Direttive* per aggiungere documentazione formattata per elementi del codice Python come funzioni, classi, moduli, ecc. Ad esempio, per documentare la funzione `io.open()`, leggendone la firma e la docstring dal file sorgente, si scrive:

```
.. autofunction:: io.open
```

Si possono anche documentare automaticamente intere classi o persino moduli, utilizzando le opzioni dei membri per le direttive automatiche, come:

```
.. automodule:: io
   :members:
```

Direttive

autodoc fornisce diverse direttive che sono versioni delle solite `py:module`, `py:class` e così via. Durante l'analisi, importano il modulo corrispondente ed estraggono la docstring degli oggetti specificati, inserendoli nel sorgente della pagina sotto una direttiva `py:module`, `py:class` ecc. adatta.

Nota: Proprio come `py:class` rispetta l'attuale `py:module`, lo farà anche `autoclass`. Allo stesso modo, `automethod` rispetterà l'attuale `py:class`.

```
.. automodule::
.. autoclass::
.. autoexception::
```

Documentare un modulo, una classe o un'eccezione. Tutte e tre le direttive inseriranno per default solo la docstring dell'oggetto stesso:

```
.. autoclass:: Noodle
```

produrrà un sorgente come questo:

```
.. class:: Noodle

    Noodle's docstring.
```

Le direttive "auto" possono anche avere un proprio contenuto, che verrà inserito nel sorgente risultante della direttiva non-auto dopo la docstring (ma prima di qualsiasi documentazione membro automatica).

Pertanto, si può anche combinare la documentazione dei membri automatica e quella non automatica, in questo modo:


```
.. autoclass:: Noodle
   :members: eat, slurp

.. method:: boil(time=10)

   Boil the noodle *time* minutes.
```

Opzioni

:members: (no value or comma separated list)

Se impostato, autodoc genererà il documento per i membri del modulo, della classe o dell'eccezione di destinazione.

Per esempio:

```
.. automodule:: noodle
   :members:
```

documenterà tutti i membri del modulo (ricorsivamente) e

```
.. autoclass:: Noodle
   :members:
```

documenterà tutti i metodi e le proprietà dei membri della classe.

Per default, autodoc non genererà documenti che siano membri privati, che non hanno docstring, ereditati dalla super classe o membri speciali.

Per i moduli, `__all__` verrà rispettato durante la ricerca dei membri a meno che non si dia l'opzione del flag `ignore-module-all`. Senza `ignore-module-all`, l'ordine dei membri sarà anche quello in `__all__`.

Si può anche fornire un elenco esplicito dei membri; solo questi verranno poi documentati:

```
.. autoclass:: Noodle
   :members: eat, slurp
```

:undoc-members: (no value)

Se impostato, autodoc genererà anche il documento per i membri che non dispongono di docstring:

```
.. automodule:: noodle
   :members:
   :undoc-members:
```

:private-members: (no value or comma separated list)

Se impostato, autodoc genererà anche la documentazione per i membri privati (ovvero quelli chiamati `_private` o `__private`):

```
.. automodule:: noodle
   :members:
   :private-members:
```

Si può anche richiedere un elenco esplicito di nomi di membri da documentare come argomenti:

```
.. automodule:: noodle
   :members:
   :private-members: _spicy, _garlickly
```

Added in version 1.1.

Cambiato nella versione 3.2: L'opzione ora può accettare argomenti.

:special-members: (no value or comma separated list)

Se impostato, autodoc genererà anche il documento per i membri speciali (cioè quelli chiamati come `__special__`):

```
.. autoclass:: my.Class
   :members:
   :special-members:
```

Si può anche richiedere un elenco esplicito di nomi di membri da documentare come argomenti:

```
.. autoclass:: my.Class
   :members:
   :special-members: __init__, __name__
```

Added in version 1.1.

Cambiato nella versione 1.2: L'opzione ora può accettare argomenti

Opzioni e utilizzo avanzato

- Per rendere di default l'opzione `members` (o altre opzioni descritte di seguito), vedere [*autodoc_default_options*](#).

Suggerimento: Si può usare una forma negata, `'no-flag'`, come opzione della direttiva autodoc, per disabilitarla temporaneamente. Per esempio:

```
.. automodule:: foo
   :no-undoc-members:
```

Suggerimento: È possibile utilizzare le opzioni della direttiva autodoc per sovrascrivere o estendere temporaneamente le opzioni di default che accettano una lista come input. Per esempio:

```
.. autoclass:: Noodle
   :members: eat
   :private-members: +_spicy, _garlickly
```

Cambiato nella versione 3.5: Le opzioni di default possono essere sovrascritte o estese temporaneamente.

- autodoc considera un membro privato se la sua docstring contiene `:meta private:` nelle sue *Liste di campi info*. Per esempio:

```
def my_function(my_arg, my_other_arg):
    """blah blah blah

    :meta private:
    """
```

Added in version 3.0.

- autodoc considera un membro pubblico se la sua docstring contiene `:meta public:` nelle sue *Liste di campi info*, anche se inizia con un'underscore. Per esempio:

```
def _my_function(my_arg, my_other_arg):
    """blah blah blah

    :meta public:
    """
```

Added in version 3.1.

- autodoc considera che un membro variabile non ha alcun valore di default se la sua docstring contiene `:meta hide-value:` nelle sue *Liste di campi info*. Esempio:

```
var1 = None #: :meta hide-value:
```

Added in version 3.5.

- Per le classi e le eccezioni, i membri ereditati dalle classi base verranno tralasciati durante la documentazione di tutti i membri, a meno che non venga fornita l'opzione `inherited-members`, oltre a `members`:

```
.. autoclass:: Noodle
   :members:
   :inherited-members:
```

Questo può essere combinato con `undoc-members` per documentare *tutti* i membri disponibili della classe o del modulo.

Può essere necessaria una classe antenata per non documentare i membri ereditati da essa. Per default, i membri della classe `object` non sono documentati. Per mostrarli tutti, si assegna `None` all'opzione.

Per esempio; Se la classe `Foo` deriva dalla classe `list` e non si vuole documentare `list.__len__()`, si dovrebbe specificare un'opzione `:inherited-members: list` per evitare membri speciali della classe `list`.

Un altro esempio; Se la classe `Foo` ha il metodo speciale `__str__` e la direttiva `autodoc` ha sia `inherited-members` che `special-members`, `__str__` sarà documentato come in passato, ma gli altri metodi speciali che non sono implementati nella classe `Foo`.

Dalla versione 5.0, può richiedere un elenco di classi antenate separate da virgole. Permette di sopprimere i membri ereditati di diverse classi sul modulo specificando contemporaneamente l'opzione alla direttiva `automodule`.

Nota: questo porterà a errori di markup se i membri ereditati provengono da un modulo le cui docstring non sono formattate reST.

Added in version 0.3.

Cambiato nella versione 3.0: Richiede il nome di una classe antenata come argomento.

Cambiato nella versione 5.0: Richiede un elenco separato da virgole di nomi di classi antenate.

- È possibile sovrascrivere (override) la firma per oggetti richiamabili esplicitamente documentati (funzioni, metodi, classi) con la sintassi regolare che sovrascriverà la firma ottenuta dall'introspezione:

```
.. autoclass:: Noodle(type)

   .. automethod:: eat(persona)
```

Questo è utile se la firma del metodo è nascosta da un decoratore.

Added in version 0.4.

- Le direttive `automodule`, `autoclass` e `autoexception` supportano anche un'opzione flag chiamata `show-inheritance`. Quando viene fornito, un elenco di classi base verrà inserito appena sotto la firma della classe (se utilizzato con `automodule`, questo verrà inserito per ogni classe documentata nel modulo).

Added in version 0.4.

- Tutte le direttive `autodoc` supportano l'opzione flag `no-index` che ha lo stesso effetto delle direttive standard `py:function` ecc.: nessuna voce di indice viene generata per l'oggetto documentato (e tutti i membri autodocumentati).

Added in version 0.4.

- `automodule` riconosce anche le opzioni `synopsis`, `platform` e `deprecated` supportate dalla direttiva standard `py:module`.

Added in version 0.5.

- `automodule` e `autoclass` hanno anche un'opzione `member-order` utilizzabile per sovrascrivere il valore globale di `autodoc_member_order` per una direttiva.

Added in version 0.6.

- Le direttive che supportano la documentazione dei membri hanno anche un'opzione `exclude-members` utilizzabile per escludere nomi di singoli membri dalla documentazione, se tutti i membri devono essere documentati.

Added in version 0.6.

- In una direttiva `automodule` con l'opzione `members` impostata, verranno documentati solo i membri del modulo il cui attributo `__module__` è uguale al nome del modulo fornito ad `automodule`. Questo per impedire la documentazione di classi e di funzioni importate. Impostare l'opzione `imported-members` per prevenire questo comportamento e documentare tutti i membri disponibili. Notare che gli attributi dei moduli importati non verranno documentati, poiché la documentazione degli attributi viene scoperta analizzando il file sorgente del modulo corrente.

Added in version 1.2.

- Aggiungere un elenco di moduli in `autodoc_mock_imports` per evitare errori di importazione che interrompano il processo di creazione quando alcune dipendenze esterne non sono importabili al momento della creazione.

Added in version 1.3.

- Come suggerimento per l'estensione `autodoc`, si può inserire un separatore `::` tra il nome del modulo e il nome dell'oggetto per consentire ad `autodoc` di conoscere il nome corretto del modulo se c'è ambiguità.

```
.. autoclass:: module.name::Noodle
```

- `autoclass` riconosce anche l'opzione `class-doc-from` utilizzabile per sovrascrivere il valore globale di `autoclass_content`.

Added in version 4.1.

```
.. autofunction::  
.. autodecorator::  
.. autodata::  
.. automethod::  
.. autoattribute::  
.. autoproperty::
```

Funzionano esattamente come `autoclass` ecc., ma non offrono le opzioni usate per la documentazione automatica dei membri.

`autodata` e `autoattribute` supportano l'opzione `annotation`. L'opzione controlla come appare il valore della variabile. Se specificato senza argomenti, verrà stampato solo il nome della variabile e il suo valore non verrà mostrato:

```
.. autodata:: CD_DRIVE
   :annotation:
```

Se l'opzione è specificata con argomenti, viene stampata dopo il nome come valore della variabile:

```
.. autodata:: CD_DRIVE
   :annotation: = your CD device name
```

Per default, senza l'opzione `annotation`, Sphinx tenta di ottenere il valore della variabile e stamparlo dopo il nome.

L'opzione `no-value` è utilizzabile al posto di una `annotation` vuota per mostrare l'hint del tipo ma non il valore:

```
.. autodata:: CD_DRIVE
   :no-value:
```

Se vengono utilizzate entrambe le opzioni `annotation` e `no-value`, `no-value` non ha alcun effetto.

Per i membri dei dati del modulo e gli attributi della classe, la documentazione può essere inserita in un commento con una formattazione speciale (usando un `#:` per iniziare il commento invece del solo `#`), o in una docstring *dopo* la definizione. I commenti devono trovarsi su una riga a parte *prima* della definizione, o immediatamente dopo l'assegnazione *sulla stessa riga*. Quest'ultima forma è limitata a una sola riga.

Ciò significa che nella seguente definizione di classe, tutti gli attributi possono essere documentati automaticamente:

```
class Foo:
    """Docstring for class Foo."""

    #: Doc comment for class attribute Foo.bar.
    #: It can have multiple lines.
    bar = 1

    flox = 1.5    #: Doc comment for Foo.flox. One line only.

    baz = 2
    """Docstring for class attribute Foo.baz."""

    def __init__(self):
        #: Doc comment for instance attribute qux.
        self.qux = 3

        self.spam = 4
        """Docstring for instance attribute spam."""
```

Cambiato nella versione 0.6: `autodata` e `autoattribute` ora possono estrarre le docstring.

Cambiato nella versione 1.1: I documenti dei commenti ora sono consentiti sulla stessa riga dopo un'assegnazione.

Cambiato nella versione 1.2: `autodata` e `autoattribute` hanno un'opzione `annotation`.

Cambiato nella versione 2.0: `autodecorator` aggiunto.

Cambiato nella versione 2.1: `autoproperty` aggiunto.

Cambiato nella versione 3.4: `autodata` e `autoattribute` ora hanno un'opzione `no-value`.

Nota: Se si documentano funzioni o metodi decorati, si tenga presente che `autodoc` recupera le sue docstring importando il modulo e controllando l'attributo `__doc__` della funzione o del metodo specificato. Questo significa che se un decoratore sostituisce la funzione decorata con un'altra, deve copiare il `__doc__` originale nella nuova funzione.

Configurazione

Ci sono anche valori di configurazione impostabili:

`autoclass_content`

Questo valore seleziona quale contenuto verrà inserito nel corpo principale di una direttiva `autoclass`. I valori possibili sono:

"class"

Viene inserita solo la docstring della classe. Questo è il default. Si può comunque documentare `__init__` come metodo separato usando `automethod` o l'opzione `members` di `autoclass`.

"both"

Sia la docstring della classe che quella del metodo `__init__` vengono concatenate e inserite.

"init"

Viene inserita solo la docstring del metodo `__init__`.

Added in version 0.3.

Se la classe non ha un metodo `__init__` o se la docstring del metodo `__init__` è vuota, ma la classe ha una docstring del metodo `__new__`, viene utilizzata al suo posto.

Added in version 1.4.

`autodoc_class_signature`

Questo valore seleziona come verrà visualizzata la firma per la classe definita dalla direttiva `autoclass`. I valori possibili sono:

"mixed"

Visualizza la firma (signature) con il nome della classe.

"separated"

Visualizza la firma come metodo.

Il default è "mixed".

Added in version 4.1.

autodoc_member_order

Questo valore seleziona se i membri documentati automaticamente sono ordinati alfabeticamente (valore 'alphabetical'), per tipologia del membro (valore 'groupwise') o per ordine del sorgente (valore 'bysource'). Il default è alphabetical.

Notare che per l'ordine del sorgente, il modulo deve essere un modulo Python con il codice sorgente disponibile.

Added in version 0.6.

Cambiato nella versione 1.0: Supporto per 'bysource'.

autodoc_default_flags

Questo valore è un elenco di flag di direttive autodoc che dovrebbero essere applicati automaticamente a tutte le direttive autodoc. I flag supportati sono 'members', 'undoc-members', 'private-members', 'special-members', 'inherited-members', 'show-inheritance', 'ignore-module-all' e 'exclude-members'.

Added in version 1.0.

Deprecato dalla versione 1.8: Integrato in [autodoc_default_options](#).

autodoc_default_options

Le opzioni di default per le direttive di autodoc. Vengono applicate automaticamente a tutte le direttive autodoc. Deve essere un dizionario che associa i nomi delle opzioni ai valori. Per esempio:

```
autodoc_default_options = {
    'members': 'var1, var2',
    'member-order': 'bysource',
    'special-members': '__init__',
    'undoc-members': True,
    'exclude-members': '__weakref__'
}
```

Impostare None o True sul valore equivale a dare solo il nome dell'opzione alle direttive.

Le opzioni supportate sono 'members', 'member-order', 'undoc-members', 'private-members', 'special-members', 'inherited-members', 'show-inheritance', 'ignore-module-all', 'imported-members', 'exclude-members', 'class-doc-from' e 'no-value'.

Added in version 1.8.

Cambiato nella versione 2.0: Accetta True come valore.

Cambiato nella versione 2.1: Aggiunto 'imported-members'.

Cambiato nella versione 4.1: Aggiunto 'class-doc-from'.

Cambiato nella versione 4.5: Aggiunto 'no-value'.

autodoc_docstring_signature

Le funzioni importate dai moduli C non possono essere analizzate e pertanto la firma di tali funzioni non può essere determinata automaticamente. Tuttavia, è una convenzione usata spesso quella di inserire la firma nella prima riga della docstring della funzione.

Se questo valore booleano è impostato su True (che è il default), autodoc esaminerà la prima riga della docstring per funzioni e metodi e, se assomiglia a una firma, utilizzerà la riga come firma e la rimuoverà dal contenuto della docstring.

autodoc continuerà a cercare più righe di firma, fermandosi alla prima riga che non sembra una firma. Ciò è utile per dichiarare firme di funzioni sovraccaricate [overloaded].

Added in version 1.1.

Cambiato nella versione 3.1: Support per le firme overloaded

Cambiato nella versione 4.0: Le firme overloaded non è necessario che siano separate da un backslash

autodoc_mock_imports

Questo valore contiene un elenco di moduli da simulare [mock up]. Questo è utile quando alcune dipendenze esterne non vengono soddisfatte in fase di creazione e interrompono il processo di building. Si può solo specificare il pacchetto root delle dipendenze stesse e omettere i sottomoduli:

```
autodoc_mock_imports = ["django"]
```

Eseguirà il mock di tutti gli import sotto il pacchetto django.

Added in version 1.3.

Cambiato nella versione 1.6: Questo valore di configurazione richiede solo la dichiarazione dei moduli di livello superiore che dovrebbero essere "mock-ati".

autodoc_typehints

Questo valore controlla come rappresentare i typehint. L'impostazione assume i seguenti valori:

- 'signature' -- Mostra i suggerimenti sul tipo (typehint) nella firma (default)
- 'description' -- Mostra i typehint come contenuto della funzione o del metodo. I typehint delle funzioni o dei metodi overloaded [sovraccaricati] saranno comunque rappresentati nella firma.
- 'none' -- Non mostra i typehint
- 'both' -- Mostra i typehint nella firma e come contenuto della funzione o del metodo

Le funzioni o i metodi overloaded [*sovraccaricati*] non avranno typehint inclusi nella descrizione perché è impossibile rappresentare con precisione tutti i possibili overload come un elenco di parametri.

Added in version 2.1.

Added in version 3.0: Aggiunta una nuova opzione 'description'.

Added in version 4.1: Aggiunta una nuova opzione 'both'.

autodoc_typehints_description_target

Questo valore controlla se i tipi di parametri non documentati e i valori restituiti sono documentati quando `autodoc_typehints` è impostato su `description`.

Il valore di default è "all", il che significa che i tipi sono documentati per tutti i parametri e i valori restituiti, indipendentemente dal fatto che siano documentati o meno.

Se impostato su "documented", i tipi verranno documentati solo per un parametro o un valore restituito che è già documentato dalla docstring.

Con "documented_params", i tipi di parametro verranno annotati solo se il parametro è documentato nella docstring. Il tipo restituito è sempre annotato (tranne se è None).

Added in version 4.0.

Added in version 5.0: È stata aggiunta la nuova opzione 'documented_params'.

autodoc_type_aliases

Un dizionario per gli utenti definito da `type aliases`²⁹⁵ che associa un nome di tipo al nome di oggetto completo. Viene utilizzato per mantenere gli alias di tipo non valutati nel documento. Il default è vuoto ({}).

Gli alias di tipo sono disponibili solo se il programma abilita la funzione **Postponed Evaluation of Annotations (PEP 563)**²⁹⁶ tramite `from __future__ import annotations`.

Per esempio, esiste un codice che utilizza un alias di tipo:

```
from __future__ import annotations

AliasType = Union[List[Dict[Tuple[int, str], Set[int]]], Tuple[str,
    ↳ List[str]]]

def f() -> AliasType:
    ...
```

Se `autodoc_type_aliases` non è impostato, autodoc genererà il markup interno da questo codice in questo modo:

```
.. py:function:: f() -> Union[List[Dict[Tuple[int, str],
    ↳ Set[int]]], Tuple[str, List[str]]]

    ...
```

Se si imposta `autodoc_type_aliases` come `{'AliasType': 'your.module.AliasType'}`, genera internamente il seguente documento:

```
.. py:function:: f() -> your.module.AliasType:
    ...
```

Added in version 3.3.

autodoc_typehints_format

Questo valore controlla il formato dei typehint. L'impostazione assume i seguenti valori:

- 'fully-qualified' -- Mostra il nome del modulo e il suo nome dei typehint
- 'short' -- Sopprime i nomi dei moduli iniziali [leading] dei typehint (es. `io.StringIO` -> `StringIO`) (il default)

Added in version 4.4.

Cambiato nella versione 5.0: L'impostazione di default è stata modificata in 'short'

autodoc_preserve_defaults

Se True, i valori degli argomenti di default delle funzioni non verranno valutati durante la generazione del documento. Li preserva così come sono nel codice sorgente.

Added in version 4.0: Aggiunta come funzionalità sperimentale. Questo verrà integrato nel core di autodoc in futuro.

autodoc_warningiserror

Questo valore controlla il comportamento di `sphinx-build -W` durante l'importazione dei moduli. Se viene fornito False, autodoc sopprime forzatamente l'errore se il modulo importato emette warning. Per default, True.

autodoc_inherit_docstrings

Questo valore controlla l'ereditarietà [inheritance] delle docstring. Se impostato su True, la docstring per classi o metodi, se non impostata esplicitamente, viene ereditata dai genitori.

Il default è True.

Added in version 1.7.

suppress_warnings

autodoc supporta la soppressione dei messaggi di warning tramite `suppress_warnings`. Permette inoltre i seguenti tipi warning:

- autodoc
- autodoc.import_object

²⁹⁵ https://mypy.readthedocs.io/en/latest/kinds_of_types.html#type-aliases

²⁹⁶ <https://peps.python.org/pep-0563/>

Preprocessing della docstring

autodoc fornisce i seguenti eventi aggiuntivi:

autodoc-process-docstring(*app, what, name, obj, options, lines*)

Added in version 0.4.

Emesso quando autodoc ha letto ed elaborato una docstring. *lines* è un elenco di stringhe - le righe della docstring elaborata - che il gestore eventi può modificare **sul posto** per cambiare ciò che Sphinx inserisce nell'output.

Parametri

- **app** -- l'oggetto dell'applicazione Sphinx
- **what** -- il tipo dell'oggetto a cui appartiene la docstring (uno tra "module", "class", "exception", "function", "method", "attribute")
- **name** -- il nome completo dell'oggetto
- **obj** -- l'oggetto stesso
- **options** -- le opzioni date alla direttiva: un oggetto con gli attributi `inherited_members`, `undoc_members`, `show_inheritance` e `no-index` che sono veri [true] se è stata data l'opzione `flag` con lo stesso nome è stata data alla direttiva `auto`
- **lines** -- le righe della docstring, vedere sopra

autodoc-before-process-signature(*app, obj, bound_method*)

Added in version 2.4.

Emesso prima che AutoDoc formatti una firma per un oggetto. Il gestore di eventi può modificare un oggetto per cambiarne la firma.

Parametri

- **app** -- l'oggetto dell'applicazione Sphinx
- **obj** -- l'oggetto stesso
- **bound_method** -- un valore booleano indica che un oggetto è bound [vincolato] o meno

autodoc-process-signature(*app, what, name, obj, options, signature, return_annotation*)

Added in version 0.5.

Emesso quando autodoc ha formattato una firma per un oggetto. Il gestore di eventi può restituire una nuova tupla (`signature`, `return_annotation`) per modificare ciò che Sphinx inserisce nell'output.

Parametri

- **app** -- l'oggetto dell'applicazione Sphinx

- **what** -- il tipo dell'oggetto a cui appartiene la docstring (uno tra "module", "class", "exception", "function", "method", "attribute")
- **name** -- il nome completo dell'oggetto
- **obj** -- l'oggetto stesso
- **options** -- le opzioni date alla direttiva: un oggetto con gli attributi `inherited_members`, `undoc_members`, `show_inheritance` e `no-index` che sono veri [true] se è stata data l'opzione flag con lo stesso nome è stata data alla direttiva `auto`
- **signature** -- firma della funzione, come una stringa nella forma "(parameter_1, parameter_2)", o None se l'introspezione non ha avuto successo e la firma non è stata specificata nella direttiva.
- **return_annotation** -- la funzione restituisce l'annotazione come stringa nella forma " -> annotation", o None se non è presente alcuna annotazione di ritorno

Il modulo `sphinx.ext.autodoc` fornisce funzioni "di fabbrica" per l'elaborazione di docstring comunemente necessaria nell'evento `autodoc-process-docstring`:

`sphinx.ext.autodoc.cut_lines`(*pre*: `int`²⁹⁷, *post*: `int`²⁹⁸ = 0, *what*: `str`²⁹⁹ | `None`³⁰⁰ = `None`) → `Callable`³⁰¹

Restituisce un listener [*ascoltatore*] che rimuove la prima *pre* e l'ultima *post* riga di ogni docstring. Se *what* è una sequenza di stringhe, verranno elaborate solo le docstring di un tipo in *what*.

Utilizzare in questo modo (ad esempio nella funzione `setup()` di `conf.py`):

```
from sphinx.ext.autodoc import cut_lines
app.connect('autodoc-process-docstring', cut_lines(4, what=['module
→ ']))
```

Questo può (e dovrebbe) essere usato al posto di `automodule_skip_lines`.

`sphinx.ext.autodoc.between`(*marker*: `str`³⁰², *what*: `Sequence`[`str`³⁰³] | `None`³⁰⁴ = `None`, *keepempty*: `bool`³⁰⁵ = `False`, *exclude*: `bool`³⁰⁶ = `False`) → `Callable`

Restituisce un listener che mantiene, o esclude se *exclude* è `True`, le righe tra quelle che corrispondono all'espressione regolare *marker*. Se nessuna riga corrisponde, la docstring risultante sarà vuota, quindi non verrà apportata alcuna modifica a meno che *keepempty* sia `true`.

Se *what* è una sequenza di stringhe, verranno elaborate solo le docstring di un tipo in *what*.

²⁹⁷ <https://docs.python.org/3/library/functions.html#int>

²⁹⁸ <https://docs.python.org/3/library/functions.html#int>

²⁹⁹ <https://docs.python.org/3/library/stdtypes.html#str>

³⁰⁰ <https://docs.python.org/3/library/constants.html#None>

³⁰¹ <https://docs.python.org/3/library/typing.html#typing.Callable>

autodoc-process-bases(*app, name, obj, options, bases*)

Emesso quando autodoc ha letto ed elaborato una classe per determinare le classi base. *bases* è un elenco di classi che il gestore di eventi può modificare **sul posto** per cambiare ciò che Sphinx inserisce nell'output. Viene emesso solo se viene fornita l'opzione `show-inheritance`.

Parametri

- **app** -- l'oggetto dell'applicazione Sphinx
- **name** -- il nome completo dell'oggetto
- **obj** -- l'oggetto stesso
- **options** -- le opzioni date alla direttiva `class`
- **bases** -- l'elenco delle firme delle classi base. vedere sopra.

Added in version 4.1.

Cambiato nella versione 4.3: *bases* può contenere una stringa come nome della classe base. Verrà elaborato come testo con markup reST.

Skipping dei membri

autodoc consente all'utente di definire un metodo personalizzato per determinare se un membro deve essere incluso nella documentazione utilizzando il seguente evento:

autodoc-skip-member(*app, what, name, obj, skip, options*)

Added in version 0.5.

Emesso quando autodoc deve decidere se un membro deve essere incluso nella documentazione. Il membro viene escluso se un handler [*gestore*] restituisce `True`. Viene incluso se l'handler restituisce `False`.

Se più di un'estensione abilitata gestisce l'evento `autodoc-skip-member`, autodoc utilizzerà il primo valore diverso da `None` restituito da un handler. Gli handler dovrebbero restituire `None` per ricadere nel comportamento di skipping di autodoc e di altre estensioni abilitate.

Parametri

- **app** -- l'oggetto dell'applicazione Sphinx
- **what** -- il tipo dell'oggetto a cui appartiene la docstring (uno tra `"module"`, `"class"`, `"exception"`, `"function"`, `"method"`, `"attribute"`)
- **name** -- il nome completo dell'oggetto

³⁰² <https://docs.python.org/3/library/stdtypes.html#str>

³⁰³ <https://docs.python.org/3/library/stdtypes.html#str>

³⁰⁴ <https://docs.python.org/3/library/constants.html#None>

³⁰⁵ <https://docs.python.org/3/library/functions.html#bool>

³⁰⁶ <https://docs.python.org/3/library/functions.html#bool>

- **obj** -- l'oggetto stesso
- **skip** -- un valore booleano che indica se autodoc salterà [skip] questo membro se l'handler dell'utente non sovrascrive la decisione.
- **options** -- le opzioni date alla direttiva: un oggetto con gli attributi `inherited_members`, `undoc_members`, `show_inheritance` e `no-index` che sono veri [true] se è stata data l'opzione flag con lo stesso nome è stata data alla direttiva `auto`

sphinx.ext.autosectionlabel -- Consente sezioni di riferimento utilizzandone il titolo

Added in version 1.4.

Questa estensione consente di fare riferimento alle sezioni con il titolo. Ciò influisce sul ruolo del riferimento (*ref*).

Per esempio:

A Plain Title

This is the text of the section.

It refers to the section title, see `:ref:`A Plain Title``.

Internamente, questa estensione genera le etichette per ciascuna sezione. Se vengono utilizzati gli stessi nomi di sezione nell'intero documento, per default viene utilizzato uno qualsiasi per un target. La variabile di configurazione `autosectionlabel_prefix_document` è utilizzabile per rendere uniche le intestazioni (titoli) che appaiono più volte ma in documenti diversi.

Configurazione

autosectionlabel_prefix_document

True per anteporre a ciascuna etichetta di sezione il nome del documento in cui si trova, seguito dai due punti. Per esempio, `index:Introduction` per una sezione chiamata `Introduction` che appare nel documento `index.rst`. Utile per evitare ambiguità quando la stessa intestazione di sezione appare in documenti diversi.

autosectionlabel_maxdepth

Se impostato, `autosectionlabel` sceglie le sezioni per l'etichettatura in base alla sua profondità. Ad esempio, quando si imposta `autosectionlabel_maxdepth` a 1, le etichette vengono generate solo per le sezioni di livello superiore e quelle più profonde non vengono etichettate. Il default è `None` (disabilitato).

Debugging

Il **WARNING: undefined label** indica che il riferimento in [ref](#) è scritto in modo errato. Invocando **sphinx-build** con `-vvv` (vedere [-v](#)³⁰⁷) verranno stampati tutti i nomi delle sezioni e le etichette generate per esse. Questo output può aiutare a trovare la giusta etichetta di riferimento.

sphinx.ext.autosummary -- Genera riepiloghi autodoc

Added in version 0.6.

Questa estensione genera elenchi di riepilogo di funzioni/metodi/attributi, simili a quelli di output, ad es. da Epydoc e altri tool di generazione di documenti API. Ciò è particolarmente utile quando le docstring sono lunghe e dettagliate e metterle ciascuna su una pagina separata ne facilita la lettura.

L'estensione `sphinx.ext.autosummary` fa questo in due parti:

1. Esiste una direttiva `autosummary` per generare elenchi di riepilogo che contengono link agli elementi documentati e brevi riepiloghi estratti dalle loro docstring.
2. Una direttiva `autosummary` genera anche brevi file "stub" per le voci elencate nel suo contenuto. Questi file per default contengono solo la direttiva `sphinx.ext.autodoc` corrispondente, ma possono essere personalizzati con i template.

Lo script **sphinx-autogen** è anche in grado di generare file "stub" dalla riga di comando.

.. autosummary::

Inserisce una tabella che contenga link a elementi documentati e un breve riepilogo (la prima frase della docstring) per ciascuno di essi.

La direttiva `autosummary` può anche opzionalmente servire come voce `toctree` per gli elementi inclusi. Facoltativamente, i file stub `.rst` per questi elementi possono anche essere generati automaticamente quando `autosummary_generate` è `True`.

Per esempio,

```
.. currentmodule:: sphinx

.. autosummary::

    environment.BuildEnvironment
    util.relative_uri
```

produce una tabella come questa:

<code>environment.</code>	L'ambiente in cui vengono tradotti i file
<code>BuildEnvironment(app)</code>	ReST.
<code>util.relative_uri(base, to)</code>	Restituisce un URL relativo da base a to.

³⁰⁷ <https://docs.python.org/3/using/cmdline.html#cmdoption-v>

Autosummary preelabora le docstring e le firme con gli stessi hook *autodoc-process-docstring* e *autodoc-process-signature* di *autodoc*.

Opzioni

- Se si vuole che la tabella *autosummary* serva anche come voce *toctree*, si usa l'opzione *toctree*, ad esempio:

```
.. autosummary::  
    :toctree: DIRNAME  
  
    sphinx.environment.BuildEnvironment  
    sphinx.util.relative_uri
```

L'opzione *toctree* segnala anche allo script **sphinx-autogen** che le pagine stub dovrebbero essere generate per le voci elencate in questa direttiva. L'opzione accetta il nome di una directory come argomento; **sphinx-autogen** inserirà per default il suo output in questa directory. Se non viene fornito alcun argomento, l'output viene inserito nella stessa directory del file che contiene la direttiva.

Si può usare anche l'opzione *caption* per dare una didascalia al toctree.

Added in version 3.1: aggiunta l'opzione della didascalia.

- Se non si vuole che *autosummary* mostri le firme [signature] delle funzioni nell'elenco, si include l'opzione *nosignatures*:

```
.. autosummary::  
    :nosignatures:  
  
    sphinx.environment.BuildEnvironment  
    sphinx.util.relative_uri
```

- Si può specificare un template custom con l'opzione *template*. Per esempio,

```
.. autosummary::  
    :template: mytemplate.rst  
  
    sphinx.environment.BuildEnvironment
```

si utilizzerebbe il template *mytemplate.rst* in *templates_path* per generare le pagine per tutte le voci elencate. Vedere *Customizing templates* di seguito.

Added in version 1.0.

- È possibile specificare l'opzione *recursive* per generare documenti per moduli e sottopacchetti in modo ricorsivo. Il default è disabled. Per esempio,

```
.. autosummary::  
    :recursive:  
  
    sphinx.environment.BuildEnvironment
```

Added in version 3.1.

sphinx-autogen -- genera pagine stub autodoc

Lo script **sphinx-autogen** è utilizzabile per generare comodamente pagine di documentazione stub per gli elementi inclusi nei listati *autosummary*.

Ad esempio, il comando

```
$ sphinx-autogen -o generated *.rst
```

leggerà tutte le tabelle *autosummary* nei file *.rst che hanno l'opzione :toctree: impostata e genererà le pagine stub corrispondenti nella directory generated per tutti gli elementi documentati. Le pagine generate per default contengono testo nel formato:

```
sphinx.util.relative_uri
=====
.. autofunction:: sphinx.util.relative_uri
```

Se l'opzione -o non viene fornita, lo script inserirà i file di output nelle directory specificate nelle opzioni :toctree:.

Per ulteriori informazioni, fare riferimento alla *documentazione di sphinx-autogen*

Generazione automatica di pagine stub

Se non si vogliono creare pagine stub con **sphinx-autogen**, si possono utilizzare questi valori di configurazione:

autosummary_context

Un dizionario di valori da passare nel contesto del motore dei template per i file stub di [autosummary].

Added in version 3.1.

autosummary_generate

Booleano che indica se scansionare tutti i documenti trovati per le direttive di autosummary e generare pagine stub per ciascuno. È abilitato per default.

Può anche essere un elenco di documenti per i quali devono essere generate le pagine stub.

I nuovi file verranno inseriti nelle directory specificate nelle opzioni :toctree: delle direttive.

Cambiato nella versione 2.3: Genera l'evento *autodoc-skip-member* come fa *autodoc*.

Cambiato nella versione 4.0: Abilitato per default.

autosummary_generate_overwrite

Se true, il riepilogo automatico sovrascrive i file esistenti mediante le pagine stub generate. Il default è true (abilitato).

Added in version 3.0.

autosummary_mock_imports

Questo valore contiene un elenco di moduli da simulare [mock up]. Consultare [autodoc_mock_imports](#) per ulteriori dettagli. Il default è [autodoc_mock_imports](#).

Added in version 2.0.

autosummary_imported_members

Un flag booleano che indica se documentare le classi e le funzioni importate nei moduli. Il default è False

Added in version 2.1.

Cambiato nella versione 4.4: Se `autosummary_ignore_module_all` è False, questo valore di configurazione viene ignorato per i membri elencati in `__all__`.

autosummary_ignore_module_all

Se False e un modulo ha l'attributo `__all__` impostato, il riepilogo automatico documenta ogni membro elencato in `__all__` e nessun altro. Il default è True

Notare che se un membro importato è elencato in `__all__`, verrà documentato indipendentemente dal valore di `autosummary_imported_members`. Per far corrispondere al comportamento di `from module import *`, impostare `autosummary_ignore_module_all` a False e `autosummary_imported_members` a True.

Added in version 4.4.

autosummary_filename_map

Un dict che associa i nomi degli oggetti ai nomi dei file. Questo è necessario per evitare conflitti di nomi di file in cui più oggetti hanno nomi indistinguibili quando viene ignorata la distinzione tra maiuscole e minuscole, su file system in cui i nomi di file non fanno distinzione.

Added in version 3.2.

Personalizzazione dei template

Added in version 1.0.

Si possono personalizzare i template della pagina stub, in modo simile ai template HTML Jinja, vedere [I template](#). ([TemplateBridge](#) non è supportato).

Nota: Se ci si ritrova a dedicare molto tempo alla personalizzazione dei template di stub, questo potrebbe indicare che è un'idea migliore scrivere invece documentazione narrativa personalizzata.

L'autosommario usa i seguenti file template Jinja:

- `autosummary/base.rst` -- template di fallback
- `autosummary/module.rst` -- template per i moduli
- `autosummary/class.rst` -- template per le classi
- `autosummary/function.rst` -- template per funzioni
- `autosummary/attribute.rst` -- template per attributi di classi
- `autosummary/method.rst` -- template per metodi di classi

Nei template sono disponibili le seguenti variabili:

name

Nome dell'oggetto documentato, esclusi il modulo e le parti della classe.

objname

Nome dell'oggetto documentato, escluse le parti del modulo.

fullname

Nome completo dell'oggetto documentato, comprese le parti del modulo e della classe.

objtype

Tipo dell'oggetto documentato, uno tra "module", "function", "class", "method", "attribute", "data", "object", "exception", "newvarattribute", "newtypedata", "property".

module

Nome del modulo a cui appartiene l'oggetto documentato.

class

Nome della classe a cui appartiene l'oggetto documentato. Disponibile solo per metodi e attributi.

underline

Una stringa contenente `len(full_name) * '='`. Utilizzare invece il filtro `underline`.

members

Elenco contenente i nomi di tutti i membri del modulo o della classe. Disponibile solo per moduli e classi.

inherited_members

Elenco contenente i nomi di tutti i membri ereditati della classe. Disponibile solo per le classi.

Added in version 1.8.0.

functions

Elenco contenente i nomi delle funzioni "public" nel modulo. Qui, "public" significa che il nome non inizia con un underscore. Disponibile solo per i moduli.

classes

Elenco contenente i nomi delle classi "public" nel modulo. Disponibile solo per i moduli.

exceptions

Elenco contenente i nomi delle eccezioni "public" nel modulo. Disponibile solo per i moduli.

methods

Elenco contenente i nomi dei metodi "public" nella classe. Disponibile solo per le classi.

attributes

Elenco contenente i nomi degli attributi "public" nella classe/modulo. Disponibile solo per classi e moduli.

Cambiato nella versione 3.1: Gli attributi dei moduli sono supportati.

modules

Elenco contenente i nomi dei moduli "public" nel pacchetto. Disponibile solo per i moduli che sono pacchetti e l'opzione `recursive` è attiva.

Added in version 3.1.

Inoltre, sono disponibili i seguenti filtri

escape(*s*)

Esegue l'*escape* di eventuali caratteri speciali nel testo da utilizzare nella formattazione dei contesti RST. Ad esempio, questo impedisce agli asterischi realizzare il grassetto. Questo sostituisce il [filtro escape](#)³⁰⁸ Jinja nativo che esegue l'escape dell'html.

underline(*s*, *line*='')

Aggiunge una sottolineatura del titolo a una parte di testo.

Per esempio, `{{ fullname | escape | underline }}` dovrebbe essere utilizzato per produrre il titolo di una pagina.

Nota: Si può usare la direttiva `autosummary` nelle pagine stub. Anche le pagine stub vengono generate in base a queste direttive.

sphinx.ext.coverage -- Raccoglie statistiche sulla copertura dei documenti

Questa estensione presenta un builder aggiuntivo, il [CoverageBuilder](#).

class sphinx.ext.coverage.CoverageBuilder

Per utilizzare questo builder, si attiva l'estensione di copertura nel file di configurazione e si inserisce `-M coverage` sulla riga di comando.

Da fare: Scrivere questa sezione.

³⁰⁸ <https://jinja.palletsprojects.com/en/3.0.x/templates/#jinja-filters.escape>

Si possono utilizzare diversi valori di configurazione per specificare cosa deve controllare il builder:

coverage_ignore_modules

coverage_ignore_functions

coverage_ignore_classes

coverage_ignore_pyobjects

Elenco delle [espressioni regolari Python](#)³⁰⁹.

Se una qualsiasi di queste espressioni regolari corrisponde a qualsiasi parte del path completo di importazione di un oggetto Python, quell'oggetto Python viene escluso dal report di copertura della documentazione.

Added in version 2.1.

coverage_c_path

coverage_c_regexes

coverage_ignore_c_items

coverage_write_headline

Impostare su False per non scrivere i titoli.

Added in version 1.1.

coverage_skip_undoc_in_source

Salta gli oggetti che non sono documentati nel sorgente con una docstring. False per default.

Added in version 1.1.

coverage_show_missing_items

Stampa anche gli oggetti omessi nell'output standard. False per default.

Added in version 3.1.

coverage_statistics_to_report

Stampare un report tabellare delle statistiche di copertura nel report di copertura. True per default.

Esempio di output:

```
+-----+-----+-----+
| Module           | Coverage | Undocumented |
+=====+=====+=====+
| package.foo_module | 100.00%  | 0            |
+-----+-----+-----+
| package.bar_module | 83.33%   | 1            |
+-----+-----+-----+
```

³⁰⁹ <https://docs.python.org/library/re>

Added in version 7.2.

coverage_statistics_to_stdout

Stampa un report tabellare delle statistiche di copertura sull'output standard. False per default.

Esempio di output:

```
+-----+-----+-----+
| Module           | Coverage | Undocumented |
+=====+=====+=====+
| package.foo_module | 100.00%  | 0            |
+-----+-----+-----+
| package.bar_module | 83.33%   | 1            |
+-----+-----+-----+
```

Added in version 7.2.

sphinx.ext.doctest -- Prova gli snippet nella documentazione

Spesso è utile includere frammenti di codice nella documentazione e mostrare i risultati della loro esecuzione. Ma è importante garantire che la documentazione sia aggiornata col codice.

Questa estensione consente di testare tali frammenti di codice nella documentazione in modo naturale. Se si contrassegnano i blocchi di codice come mostrato qui, il builder `doctest` li raccoglierà e li eseguirà come test `doctest`.

All'interno di ogni documento, si può assegnare ogni snippet a un *gruppo*. Ogni gruppo è composto da:

- zero o più blocchi di *codice di setup* (ad esempio importando il modulo da testare)
- uno o più blocchi di *test*

Quando si creano documenti con il builder `doctest`, i gruppi vengono raccolti per ciascun documento ed eseguiti uno dopo l'altro, eseguendo prima i blocchi di codice di setup, poi i blocchi di test nell'ordine in cui appaiono nel file.

Esistono due tipi di blocchi di test:

- I blocchi *doctest-style* imitano le sessioni interattive interlacciando il codice Python (incluso il prompt dell'interprete) con l'output.
- I blocchi *code-output-style* sono costituiti da un normale pezzo di codice Python e, facoltativamente, da un pezzo di output per quel codice.

Direttive

L'argomento *group* riportato di seguito viene interpretato come segue: se è vuoto, il blocco viene assegnato al gruppo denominato `default`. Se è `*`, il blocco viene assegnato a tutti i gruppi (incluso il gruppo `default`). Altrimenti, deve essere un elenco di nomi di gruppi separati da virgole.

`.. testsetup:: [group]`

Un blocco di codice di configurazione. Questo codice non viene mostrato nell'output per altri builder, ma viene eseguito prima dei doctest dei gruppi a cui appartiene.

`.. testcleanup:: [group]`

Un blocco di codice di pulizia [cleanup]. Questo codice non viene mostrato nell'output per altri builder, ma eseguito dopo i doctest dei gruppi a cui appartiene.

Added in version 1.1.

`.. doctest:: [group]`

Un blocco di codice in stile doctest. Si possono usare i flag standard `doctest`³¹⁰ per controllare come l'output effettivo viene confrontato con quello che si fornisce come output. Il set di default dei flag è specificato dalla variabile di configurazione `doctest_default_flags`.

Questa direttiva supporta cinque opzioni:

- `hide`, un'opzione flag, nasconde il blocco doctest in altri builder. Per default viene mostrato come un blocco doctest evidenziato.
- `options`, un'opzione stringa, può essere utilizzata per fornire un elenco separato da virgole di flag doctest che si applicano a ciascun esempio nei test. (Si possono comunque fornire flag espliciti per esempio, con commenti doctest, ma verranno visualizzati anche in altri builder).
- `pyversion`, un'opzione stringa, utilizzabile per specificare la versione Python richiesta per l'esempio da testare. Ad esempio, nel caso seguente l'esempio verrà testato solo per versioni Python successive alla 3.10:

```
.. doctest::  
   :pyversion: > 3.10
```

Sono supportati i seguenti operandi:

- `~=`: Clausola per la release compatibile
- `==`: Clausola per la versione corrispondente
- `!=`: Clausola per l'esclusione della versione
- `<=`, `>=`: Clausola inclusiva per il confronto ordinato
- `<`, `>`: Clausola esclusiva per il confronto ordinato
- `===`: Clausola arbitraria di uguaglianza.

L'opzione `pyversion` è seguita **PEP-440: Version Specifiers**³¹¹.

Added in version 1.6.

Cambiato nella versione 1.7: Operandi e notazioni PEP-440 supportate

- `trim-doctest-flags` e `no-trim-doctest-flags`, un'opzione di flag, flag `doctest` (commenti simili a `# doctest: FLAG, ...`) alla fine delle righe e i marcatori `<BLANKLINE>` vengono rimossi (o non rimossi) individualmente. Il default è `trim-doctest-flags`.

Notare che come con i `doctest` standard, si deve usare `<BLANKLINE>` per segnalare una riga vuota nell'output previsto. Il `<BLANKLINE>` viene rimosso durante la creazione [building] dell'output della presentazione (HTML, LaTeX ecc.).

Inoltre, si possono fornire opzioni di `doctest` in linea, come in `doctest`:

```
>>> datetime.date.now()    # doctest: +SKIP
datetime.date(2008, 1, 1)
```

Saranno rispettati durante l'esecuzione del test, ma rimossi dall'output della presentazione.

.. **testcode::** [group]

Un blocco di codice per un test in stile code-output.

Questa direttiva supporta tre opzioni:

- `hide`, un'opzione flag, nasconde il blocco di codice in altri builder. Per default viene mostrato come un blocco di codice evidenziato.
- `trim-doctest-flags` e `no-trim-doctest-flags`, un'opzione di flag, flag `doctest` (commenti simili a `# doctest: FLAG, ...`) alla fine delle righe e i marcatori `<BLANKLINE>` vengono rimossi (o non rimossi) individualmente. Il default è `trim-doctest-flags`.

Nota: Il codice in un blocco `testcode` viene sempre eseguito tutto in una volta, non importa quante istruzioni contenga. Pertanto, l'output *non* verrà generato per le semplici espressioni -- utilizzare `print`. Esempio:

```
.. testcode::

    1+1          # this will give no output!
    print(2+2)   # this will give output

.. testoutput::

    4
```

³¹⁰ <https://docs.python.org/3/library/doctest.html#module-doctest>

³¹¹ <https://peps.python.org/pep-0440/#version-specifiers>

Inoltre, si tenga presente che poiché il modulo `doctest` non supporta la combinazione di output regolare e un messaggio di eccezione nello stesso snippet, ciò si applica anche a `testcode/testoutput`.

.. testoutput:: [group]

L'output corrispondente, o il messaggio di eccezione, per l'ultimo blocco `testcode`.

Questa direttiva supporta quattro opzioni:

- `hide`, un'opzione flag, nasconde il blocco di output in altri builder. Per default viene visualizzato come un blocco letterale senza evidenziazione.
- `options`, un'opzione stringa, può essere utilizzata per fornire flag `doctest` (separati da virgole) proprio come nei normali blocchi `doctest`.
- `trim-doctest-flags` e `no-trim-doctest-flags`, un'opzione di flag, flag `doctest` (commenti simili a `# doctest: FLAG, ...`) alla fine delle righe e i marcatori `<BLANKLINE>` vengono rimossi (o non rimossi) individualmente. Il default è `trim-doctest-flags`.

Esempio:

```
.. testcode::

    print('Output      text.')
```

```
.. testoutput::
:hide:
:options: -ELLIPSIS, +NORMALIZE_WHITESPACE

Output text.
```

Quello che segue è un esempio di utilizzo delle direttive. Il test tramite `doctest` e quello tramite `testcode` e `testoutput` sono equivalenti.

The parrot module

```
.. testsetup:: *
```

```
import parrot
```

The parrot module is a module about parrots.

Doctest example:

```
.. doctest::
```

```
>>> parrot.voom(3000)
```

```
This parrot wouldn't voom if you put 3000 volts through it!
```

(continues on next page)

(continua dalla pagina precedente)

Test-Output example:

```
.. testcode::
```

```
    parrot.voom(3000)
```

This would output:

```
.. testoutput::
```

```
    This parrot wouldn't voom if you put 3000 volts through it!
```

Saltare i test in modo condizionale

`skipif`, un'opzione stringa, può essere utilizzata per saltare le direttive in modo condizionale. Questo può essere utile ad es. quando è necessario eseguire una serie diversa di test a seconda dell'ambiente (hardware, rete/VPN, dipendenze opzionali o versioni diverse delle dipendenze). L'opzione `skipif` è supportata da tutte le direttive doctest. Di seguito sono riportati i casi d'uso tipici di `skipif` quando utilizzato per direttive diverse:

- `testsetup` e `testcleanup`
 - salta in modo condizionale la configurazione e/o la pulizia [cleanup] del test
 - personalizza il codice di installazione/cleanup per ambiente
- `doctest`
 - salta condizionalmente sia un test che la relativa verifica dell'output
- `testcode`
 - salta condizionalmente un test
 - personalizza il codice di test per ambiente
- `testoutput`
 - salta condizionalmente l'asserzione di output per un test saltato
 - si aspetta risultati diversi a seconda dell'ambiente

Il valore dell'opzione `skipif` viene valutato come un'espressione Python. Se il risultato è un valore `true`, la direttiva viene omessa dall'esecuzione del test proprio come se non fosse affatto presente nel file.

Invece di ripetere un'espressione, l'opzione di configurazione `doctest_global_setup` può essere utilizzata per assegnarla a una variabile che può poi essere utilizzata al suo posto.

Ecco un esempio che salta alcuni test se Pandas non è installato:

Listato 1: conf.py

```
extensions = ['sphinx.ext.doctest']
doctest_global_setup = '''
try:
    import pandas as pd
except ImportError:
    pd = None
'''
```

Listato 2: contents.rst

```
.. testsetup::
   :skipif: pd is None

   data = pd.Series([42])

.. doctest::
   :skipif: pd is None

   >>> data.iloc[0]
   42

.. testcode::
   :skipif: pd is None

   print(data.iloc[-1])

.. testoutput::
   :skipif: pd is None

   42
```

Configurazione

L'estensione doctest utilizza i seguenti valori di configurazione:

doctest_default_flags

Per default, queste opzioni sono abilitate:

- **ELLIPSIS**, che consente di inserire dei puntini di sospensione nell'output previsto che corrispondono a qualsiasi cosa nell'output effettivo;
- **IGNORE_EXCEPTION_DETAIL**, fa sì che tutto ciò che segue i due punti più a sinistra e qualsiasi informazione sul modulo nel nome dell'eccezione venga ignorato;

- `DONT_ACCEPT_TRUE_FOR_1`, rifiuta "True" nell'output dove viene fornito "1" - il comportamento di default di accettare questa sostituzione è una reliquia di pre-Python 2.2.

Added in version 1.5.

doctest_show_successes

Il default è `True`. Controlla se vengono riportati i successi.

Per un progetto con molti doctest, potrebbe essere utile impostarlo su `False` per evidenziare solo gli errori.

Added in version 7.2.

doctest_path

Un elenco di directory che verranno aggiunte a `sys.path`³¹² quando viene utilizzato il builder doctest. (Deve contenere path assoluti).

doctest_global_setup

Codice Python che viene trattato come se fosse inserito in una direttiva `testsetup` per *ciascun* file testato e per ogni gruppo. Lo si può usare, per esempio, per importare moduli che saranno sempre necessari nei propri doctest.

Added in version 0.6.

doctest_global_cleanup

Codice Python che viene trattato come se fosse inserito in una direttiva `testcleanup` per *ciascun* file testato e per ogni gruppo. Lo si può usare, per esempio, per rimuovere eventuali file temporanei lasciati dai test.

Added in version 1.1.

doctest_test_doctest_blocks

Se questa è una stringa non vuota (il valore predefinito è `'default'`), verranno testati anche i blocchi doctest standard di reST. Verranno assegnati al nome del gruppo indicato.

I blocchi doctest reST sono semplicemente dei doctest inseriti in un paragrafo a parte, in questo modo:

```
Some documentation text.

>>> print(1)
1

Some more documentation text.
```

(Notare che non viene utilizzato alcun `::` speciale per introdurre un blocco doctest; docutils li riconosce dal `>>>` iniziale. Inoltre, non viene utilizzata alcuna indentazione aggiuntiva, anche se non nuoce).

Se questo valore viene lasciato al suo valore di default, il frammento di cui sopra viene interpretato dal builder doctest esattamente come segue:

³¹² <https://docs.python.org/3/library/sys.html#sys.path>

```
Some documentation text.

.. doctest::

    >>> print(1)
    1

Some more documentation text.
```

Questa funzionalità semplifica il test dei doctest nelle docstring incluse con l'estensione *autodoc* senza contrassegnarli con una direttiva speciale.

Notare, tuttavia, che non si possono avere righe vuote nei blocchi doctest reST. Verranno interpretati come la fine di un blocco e l'inizio di un altro.. Inoltre, la rimozione delle opzioni `<BLANKLINE>` e `# doctest:` funziona solo nei blocchi *doctest*, anche se è possibile impostare *trim_doctest_flags* per ottenere ciò in tutti i blocchi di codice con il contenuto della console Python.

sphinx.ext.duration -- Misura la durata dell'elaborazione di Sphinx

Added in version 2.4.

Questa estensione misura la durata dell'elaborazione di Sphinx e ne mostra il risultato alla fine della build. È utile per verificare quale documento viene "buildato" lentamente.

sphinx.ext.extlinks -- Markup per abbreviare i link esterni

Autore del modulo: Georg Brandl

Added in version 1.0.

Questa estensione ha lo scopo di aiutare con il pattern comune di avere molti link esterni che puntano a URL su uno stesso sito, ad es. link a bug tracker, interfacce web di controllo della versione o semplicemente sottopagine in altri siti web. Lo fa fornendo alias agli URL di base, in modo che si debba solo fornire il nome della pagina secondaria quando si crea un link.

Supponiamo che si vogliano includere molti link a problemi nel tracker Sphinx, su <https://github.com/sphinx-doc/sphinx/issues/num>. Digitare questo URL ripetutamente è noioso, quindi si può usare *extlinks* per evitare di ripeterli.

L'estensione aggiunge un valore di configurazione:

extlinks

Questo valore di configurazione deve essere un dizionario di siti esterni, che associa nomi alias brevi univoci a un *URL base* e una *didascalia*. Ad esempio, per creare un alias per i problemi sopra menzionati, si dovrebbe aggiungere

```
extlinks = {'issue': ('https://github.com/sphinx-doc/sphinx/issues/
→%s',
                    'issue %s')}
```

Ora si può utilizzare l'alias del nome come nuovo ruolo, ad es. `:issue:`123``. Questo inserisce poi un link a <https://github.com/sphinx-doc/sphinx/issues/123>. Come si vede, l'obiettivo fornito nel ruolo viene sostituito nello*URL base* al posto di %s.

La didascalia del link dipende dal secondo elemento nella tupla, la *caption*:

- Se *caption* è None, la didascalia del link è l'URL completo.
- Se *caption* è una stringa, allora deve contenere %s esattamente una volta. In questo caso la didascalia del link è *caption* con l'URL parziale sostituito da %s -- nell'esempio sopra, la didascalia del link sarebbe `issue 123`.

Per produrre un % letterale nello *URL base* o in *caption*, si usa %%:

```
extlinks = {'KnR': ('https://example.org/K%%26R/page/%s',
                    '[K&R; page %s]')}
```

Si può anche utilizzare la consueta sintassi del "titolo esplicito" supportata da altri ruoli che generano link, ad esempio `:issue:`this issue <123>``. In questo caso, *caption* non è rilevante.

Cambiato nella versione 4.0: Supporto per sostituire con '%s' nella didascalia.

Nota: Poiché i link vengono generati dal ruolo nella fase di lettura, appaiono come link ordinari ad es. il builder `linkcheck`.

extlinks_detect_hardcoded_links

Se abilitato, `extlinks` emette un warning se un link hardcoded è sostituibile da un extlink e suggerisce una sostituzione tramite warning. Il default è `False`.

Added in version 4.5.

sphinx.ext.githubpages -- Pubblica documenti HTML nelle pagine GitHub

Added in version 1.4.

Cambiato nella versione 2.0: Supporta il file `CNAME`

Questa estensione crea il file `.nojekyll` nella directory HTML generata per pubblicare il documento su GitHub Pages.

Crea anche un file `CNAME` per i domini personalizzati quando è impostato `html_baseurl`.

sphinx.ext.graphviz -- Aggiunge grafici Graphviz

Added in version 0.6.

Questa estensione ti consente di incorporare grafici [Graphviz](https://graphviz.org/)³¹³ nei documenti.

Aggiunge queste direttive:

.. graphviz::

Direttiva per incorporare il codice graphviz. Il codice di input per dot viene fornito come contenuto. Per esempio:

```
.. graphviz::  
  
    digraph foo {  
        "bar" -> "baz";  
    }
```

Nell'output HTML, il codice verrà reso in un'immagine PNG o SVG (vedere [graphviz_output_format](#)). Nell'output LaTeX, il codice verrà reso in un file PDF incorporabile.

Si possono anche incorporare file dot esterni, fornendo il nome del file come argomento a [graphviz](#) e nessun contenuto aggiuntivo:

```
.. graphviz:: external.dot
```

Come per tutti i riferimenti a file in Sphinx, se il nome del file è assoluto, viene considerato relativo alla directory sorgente.

Cambiato nella versione 1.1: Aggiunto il supporto per i file esterni.

opzioni

:alt: alternate text (text)

Il testo alternativo del grafico. Per default, il codice del grafico viene utilizzato per il testo alternativo.

Added in version 1.0.

:align: alignment of the graph (left, center or right)

L'allineamento orizzontale del grafico.

Added in version 1.5.

:caption: caption of the graph (text)

La didascalia del grafico.

Added in version 1.1.

³¹³ <https://graphviz.org/>

:layout: layout type of the graph (text)

Il layout del grafico (es. dot, neato e così via). È consentito anche un path ai comandi graphviz. Per default, viene usato *graphviz_dot*.

Added in version 1.4.

Cambiato nella versione 2.2: Rinominato da graphviz_dot

:name: label (text)

L'etichetta del grafico.

Added in version 1.6.

:class: class names (a list of class names separated by spaces)

Il nome della classe del grafico.

Added in version 2.4.

.. graph::

Direttiva per l'inclusione di un singolo grafico non orientato. Il nome viene fornito come argomento della direttiva, il contenuto del grafico è il contenuto della direttiva. Questa è una direttiva di convenienza per generare graph <name> { <content> }.

Per esempio:

```
.. graph:: foo

    "bar" -- "baz";
```

Nota: Il nome del grafico viene passato invariato a Graphviz. Se contiene caratteri non alfanumerici (ad esempio un trattino), lo si dovrà mettere tra virgolette doppie.

opzioni

Uguale a *graphviz*.

:alt: alternate text (text)

Added in version 1.0.

:align: alignment of the graph (left, center or right)

Added in version 1.5.

:caption: caption of the graph (text)

Added in version 1.1.

:layout: layout type of the graph (text)

Added in version 1.4.

Cambiato nella versione 2.2: Rinominato da graphviz_dot

:name: label (text)

Added in version 1.6.

:class: class names (a list of class names separated by spaces)

Il nome della classe del grafico.

Added in version 2.4.

.. digraph::

Direttiva per incorporare un singolo grafico diretto. Il nome viene fornito come argomento della direttiva, il contenuto del grafico è il contenuto della direttiva. Questa è una direttiva di convenienza per generare `digraph <name> { <content> }`.

Per esempio:

```
.. digraph:: foo

    "bar" -> "baz" -> "quux";
```

opzioni

Uguale a *graphviz*.

:alt: alternate text (text)

Added in version 1.0.

:align: alignment of the graph (left, center or right)

Added in version 1.5.

:caption: caption of the graph (text)

Added in version 1.1.

:layout: layout type of the graph (text)

Added in version 1.4.

Cambiato nella versione 2.2: Rinominato da `graphviz_dot`

:name: label (text)

Added in version 1.6.

:class: class names (a list of class names separated by spaces)

Il nome della classe del grafico.

Added in version 2.4.

Ci sono anche questi valori di configurazione:

graphviz_dot

Il nome del comando con cui invocare dot. Il default è 'dot'; potrebbe essere necessario impostarlo su un path completo se dot non è presente nel path di ricerca degli eseguibili.

Dato che questa impostazione non è trasferibile da un sistema all'altro, normalmente non è utile impostarla in `conf.py`; piuttosto, dovrebbe essere preferibile fornirlo sulla riga di comando **sphinx-build** tramite l'opzione `-D`, in questo modo:

```
sphinx-build -M html -D graphviz_dot=C:\graphviz\bin\dot.exe . _
↪build
```

graphviz_dot_args

Argomenti aggiuntivi della riga di comando da fornire a dot, come lista. Il default è una lista vuota. Questo è il posto giusto per impostare gli attributi globali del grafico, del nodo o del bordo tramite le opzioni `-G`, `-N` e `-E` di dot.

graphviz_output_format

Il formato di output per Graphviz durante la creazione di file HTML. Deve essere `'png'` o `'svg'`; il default è `'png'`. Se viene utilizzato `'svg'`, per far funzionare correttamente i link URL, è necessario impostare un attributo `target` appropriato, come `"_top"` e `"_blank"`. Ad esempio, il link nel grafico seguente dovrebbe funzionare nell'output `svg`:

```
.. graphviz::

    digraph example {
        a [label="sphinx", href="https://www.sphinx-doc.org/", ↪
↪target="_top"];
        b [label="other"];
        a -> b;
    }
```

Added in version 1.0: In precedenza, l'output era sempre PNG.

sphinx.ext.ifconfig -- Include il contenuto in base alla configurazione

Questa estensione è abbastanza semplice e presenta una sola direttiva:

Avvertimento: Questa direttiva è progettata per controllare solo il contenuto del documento. Non poteva controllare sezioni, etichette e così via.

.. ifconfig::

Include il contenuto della direttiva solo se l'espressione Python fornita come argomento è `True`, valutata nello spazio dei nomi della configurazione del progetto (ovvero, tutte le variabili registrate da `conf.py` sono disponibili).

Ad esempio, si potrebbe scrivere:

```
.. ifconfig:: releaselevel in ('alpha', 'beta', 'rc')

    This stuff is only included in the built docs for unstable ↪
↪versions.
```

Per comunicare a Sphinx un valore di configurazione personalizzato, utilizzare `add_config_value()` nella funzione di configurazione in `conf.py`, ad esempio:

```
def setup(app):
    app.add_config_value('releaselevel', '', 'env')
```

Il secondo argomento è il valore di default, il terzo dovrebbe sempre essere `'env'` per tali valori (seleziona se Sphinx rilegge i documenti se il valore cambia).

sphinx.ext.imgconverter -- Un convertitore di immagini di riferimento che utilizza Imagemagick

Added in version 1.6.

Questa estensione converte le immagini nel documento in un formato appropriato per i builder. Ad esempio, consente di utilizzare immagini SVG con il builder LaTeX. Di conseguenza, non importa quale formato dell'immagine supporta il builder.

Per default, l'estensione utilizza [ImageMagick](https://www.imagemagick.org)³¹⁴ per eseguire le conversioni e non funzionerà se ImageMagick non è installato.

Nota: ImageMagick rasterizza un'immagine SVG durante la conversione. Di conseguenza, l'immagine diventa non scalabile. Per evitare ciò, utilizza altri convertitori di immagini come [sphinxcontrib-svg2pdfconverter](https://github.com/missinglinkelectronics/sphinxcontrib-svg2pdfconverter)³¹⁵ (che usa Inkscape o `rsvg-convert`).

Configurazione

image_converter

Un path per un comando di conversione. Per default, `imgconverter` trova il comando dai path di ricerca.

Sulle piattaforme Unix, il comando **convert** viene utilizzato per default.

Su Windows, il comando **magick** viene usato per default.

Cambiato nella versione 3.1: Usa il comando **magick** per default su windows

image_converter_args

Argomenti aggiuntivi della riga di comando da fornire a **convert**, come una list. Il default è una lista vuota `[]`.

Su Windows, il default è `["convert"]`.

Cambiato nella versione 3.1: Usa `["convert"]` per default su Windows

³¹⁴ <https://www.imagemagick.org>

³¹⁵ <https://github.com/missinglinkelectronics/sphinxcontrib-svg2pdfconverter>

sphinx.ext.inheritance_diagram -- Include diagrammi di ereditarietà

Added in version 0.6.

Questa estensione consente di includere diagrammi di ereditarietà [inheritance diagrams], resi tramite l'estensione [Graphviz extension](#).

Aggiunge questa direttiva:

.. inheritance-diagram::

Questa direttiva ha uno o più argomenti, ciascuno dei quali fornisce un nome di modulo o di classe. I nomi delle classi possono essere senza qualifica; in tal caso si considera che esistano nel modulo attualmente descritto (vedere [py:module](#)).

Per ciascuna classe e per ciascuna classe in ciascun modulo vengono determinate le classi base. Poi, da tutte le classi e dalle relative classi base, viene generato un grafico che viene poi rappresentato tramite l'estensione graphviz in un grafo diretto.

Questa direttiva supporta un'opzione chiamata "parts" che, se fornita, deve essere un numero intero, consigliando alla direttiva di mantenere un certo numero di parti separate da punti nei nomi visualizzati (da destra a sinistra). Ad esempio, `parts=1` mostrerà solo i nomi delle classi, senza quelli dei moduli che le contengono.

Cambiato nella versione 2.0: Il valore di `parts` può anche essere negativo, indicando quante parti eliminare da sinistra. Ad esempio, se tutti i nomi delle classi iniziano con `lib.`, si può fornire `:parts: -1` per rimuovere tale prefisso dai nomi dei nodi visualizzati.

La direttiva supporta anche l'opzione flag `private-bases` se specificato, verranno incluse le classi base private (quelle il cui nome inizia con ```_`).

Si può usare l'opzione `caption` per dare una didascalia al diagramma.

Cambiato nella versione 1.1: Aggiunta l'opzione `private-bases`; in precedenza, tutte le basi venivano sempre incluse.

Cambiato nella versione 1.5: Aggiunta l'opzione `caption`

Supporta anche l'opzione `top-classes` che richiede uno o più nomi di classi separati da virgola. Se specificato, l'attraversamento dell'ereditarietà si fermerà ai nomi delle classi specificate. Dato il seguente modulo Python:

```

"""
    A
   /\
  B  C
 /\  /\
E  D  F
"""

class A:
    pass

```

(continues on next page)

(continua dalla pagina precedente)

```
class B(A):
    pass

class C(A):
    pass

class D(B, C):
    pass

class E(B):
    pass

class F(C):
    pass
```

Se è stato specificato un modulo nel diagramma di ereditarietà in questo modo:

```
.. inheritance-diagram:: dummy.test
   :top-classes: dummy.test.B, dummy.test.C
```

tutte le classi base antenate delle `top-classes` e che sono anche definite nello stesso modulo verranno rese come nodi autonomi [alone]. In questo esempio la classe A verrà resa come nodo autonomo nel grafico. Si tratta di un problema noto dovuto al funzionamento interno di questa estensione.

Se non si vuole che la classe A (o qualsiasi altro antenato) sia visibile, si specificano solo le classi per le quali si vogliono generare il diagramma in questo modo:

```
.. inheritance-diagram:: dummy.test.D dummy.test.E dummy.test.F
   :top-classes: dummy.test.B, dummy.test.C
```

Cambiato nella versione 1.7: Aggiunta l'opzione `top-classes` per limitare lo scope dei grafici di ereditarietà.

Esempi

Quelli che seguono sono diversi diagrammi di ereditarietà per la classe interna `InheritanceDiagram` che implementa la direttiva.

Con nomi completi:

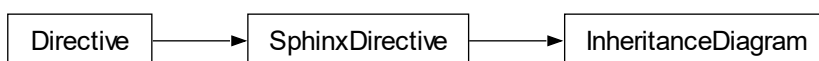
```
.. inheritance-diagram:: sphinx.ext.inheritance_diagram.
   ↳ InheritanceDiagram
```



Mostra solo i nomi delle classi:

```

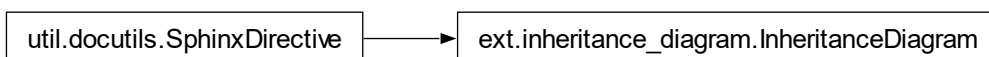
.. inheritance-diagram:: sphinx.ext.inheritance_diagram.
   ↳ InheritanceDiagram
   :parts: 1
  
```



Arrestando il diagramma in *sphinx.util.docutils.SphinxDirective* (la superclasse più alta che fa ancora parte di Sphinx) ed eliminando la parte comune più a sinistra (sphinx) da tutti i nomi:

```

.. inheritance-diagram:: sphinx.ext.inheritance_diagram.
   ↳ InheritanceDiagram
   :top-classes: sphinx.util.docutils.SphinxDirective
   :parts: -1
  
```



class sphinx.ext.inheritance_diagram.InheritanceDiagram

La classe interna che implementa la direttiva inheritance-diagram.

Configurazione

`inheritance_graph_attrs`

Un dizionario di attributi del grafico graphviz per i diagrammi di ereditarietà.

Per esempio:

```
inheritance_graph_attrs = dict(rankdir="LR", size='"6.0, 8.0"',
                               fontsize=14, ratio='compress')
```

`inheritance_node_attrs`

Un dizionario degli attributi dei nodi graphviz per i diagrammi di ereditarietà.

Per esempio:

```
inheritance_node_attrs = dict(shape='ellipse', fontsize=14,
                               ↪height=0.75,
                               color='dodgerblue1', style='filled')
```

`inheritance_edge_attrs`

Un dizionario di attributi dei bordi graphviz per i diagrammi di ereditarietà.

`inheritance_alias`

Consente di mappare il nome completo e qualificato della classe su valori personalizzati (utile quando non è auspicabile esporre il path sottostante di una classe, ad esempio è una classe privata e non deve essere istanziata dall'utente).

Per esempio:

```
inheritance_alias = {'_pytest.Magic': 'pytest.Magic'}
```

`sphinx.ext.intersphinx` -- [Link alla documentazione di altri progetti](#)

Added in version 0.5.

Questa estensione può generare link alla documentazione di oggetti in progetti esterni, sia esplicitamente attraverso il ruolo `external`, sia come risoluzione di fallback per qualsiasi altro riferimento incrociato.

L'utilizzo per la risoluzione del fallback è semplice: ogni volta che Sphinx incontra un riferimento incrociato che non ha una target corrispondente nel set di documentazione corrente, li cerca nei set di documentazione esterna configurati in `intersphinx_mapping`. Un riferimento come `:py:class:`zipfile.ZipFile`` può quindi collegarsi alla documentazione Python per la classe `ZipFile`, senza che si debba specificare dove si trova esattamente.

Quando si usa il ruolo `external`, si può forzare la ricerca su qualsiasi progetto esterno e, facoltativamente, su un progetto esterno specifico. Un link come `:external:ref:`comparison manual <comparisons>`` si collegherà quindi all'etichetta "comparisons" in qualunque progetto esterno configurato, se esiste, e un link come `:external+python:ref:`comparison`

`manual <comparisons>`` collegherà all'etichetta "comparisons" solo nel set di documenti "python", se esiste.

Dietro le quinte, funziona come segue:

- Ogni build HTML di Sphinx crea un file denominato `objects.inv` che contiene una mappatura dai nomi degli oggetti agli URI relativi alla radice del set HTML.
- I progetti che utilizzano l'estensione Intersphinx possono specificare la posizione di tali file di mappatura nel valore di configurazione `intersphinx_mapping`. La mappatura verrà quindi utilizzata per risolvere sia i riferimenti `external`, sia i riferimenti altrimenti mancanti agli oggetti in collegamenti all'altra documentazione.
- Per default, si presuppone che il file di mappatura si trovi nella stessa posizione del resto della documentazione; tuttavia, tale posizione può essere specificata anche individualmente, ad es. se i documenti dovessero essere compilabili senza accesso a Internet.

Configurazione

Per utilizzare il link Intersphinx, si aggiunge `'sphinx.ext.intersphinx'` al valore di configurazione `extensions` e utilizza tali valori di configurazione per attivare il collegamento:

`intersphinx_mapping`

Questo valore di configurazione contiene le posizioni e i nomi di altri progetti che dovrebbero essere collegati in questa documentazione.

I path locali relativi per le posizioni di destinazione vengono considerati relativi alla base della documentazione creata, mentre i path locali relativi per le posizioni di inventario vengono considerati relativi alla directory di origine.

Quando si recuperano [fetch] file di inventario remoti, le impostazioni del proxy verranno lette dalla variabile di ambiente `$HTTP_PROXY`.

Formato

Added in version 1.0.

Un dizionario che mappa identificatori univoci su una tupla (`target`, `inventory`). Ogni `target` è l'URI di base di un set di documentazione Sphinx esterno e può essere un path locale o un URI HTTP. `inventory` indica dove è possibile trovare il file di inventario: può essere `None` (un file `objects.inv` nella stessa posizione dell'URI di base) o un altro path di file locale o un URI HTTP completo in un file di inventario.

L'identificatore univoco può essere utilizzato nel ruolo `external`, in modo che sia chiaro a quale set intersphinx appartiene il target. Un link come `external:python+ref:`comparison manual <comparisons>`` si collegherà all'etichetta "comparisons" nel set di documenti "python", se esiste.

Esempio

Per aggiungere link a moduli e oggetti nella documentazione della libreria standard Python, utilizzare:

```
intersphinx_mapping = {'python': ('https://docs.python.org/3',  
→None)}
```

Questo scaricherà il file `objects.inv` corrispondente da Internet e genererà link alle pagine sotto l'URI specificato. L'inventario scaricato viene memorizzato nella cache dell'ambiente Sphinx, quindi deve essere scaricato nuovamente ogni volta che si riesegue una build completa.

Un secondo esempio, che mostra il significato di un valore diverso da `None` del secondo elemento della tupla:

```
intersphinx_mapping = {'python': ('https://docs.python.org/3',  
                                'python-inv.txt')}
```

Questo leggerà l'inventario da `python-inv.txt` nella directory sorgente, ma genererà comunque collegamenti alle pagine in `https://docs.python.org/3`. Il file di inventario va aggiornato manualmente man mano che nuovi oggetti vengono aggiunti alla documentazione Python.

Target multipli per l'inventario

Added in version 1.3.

È possibile specificare file alternativi per ciascun inventario. È possibile fornire una tupla per il secondo elemento tupla dell'inventario, come mostrato nell'esempio seguente. Questo leggerà l'inventario scorrendo i (secondi) elementi della tupla fino al primo recupero riuscito. Il motivo principale è quello di poter specificare i siti mirror per i tempi di inattività del server dell'inventario primario:

```
intersphinx_mapping = {'python': ('https://docs.python.org/3',  
                                (None, 'python-inv.txt'))}
```

Per una serie di libri modificati e testati localmente e poi pubblicati insieme, potrebbe essere utile provare prima un file di inventario locale, per controllare i riferimenti prima della pubblicazione:

```
intersphinx_mapping = {  
    'otherbook':  
        ('https://myproj.readthedocs.io/projects/otherbook/en/  
→latest',  
         ('../otherbook/build/html/objects.inv', None)),  
}
```

Vecchio formato per questo valore di configurazione

Deprecato dalla versione 6.2.

Attenzione: Questo è il formato utilizzato prima di Sphinx 1.0. È deprecato e verrà rimosso in Sphinx 8.0.

Un dizionario che mappa gli URI su `None` o su un URI. Le chiavi sono l'URI di base dei set di documentazione Sphinx esterni e possono essere path locali o URI HTTP. I valori indicano dove è possibile trovare il file di inventario: possono essere `None` (nella stessa posizione dell'URI di base) o un altro URI locale o HTTP.

Esempio:

```
intersphinx_mapping = {'https://docs.python.org/': None}
```

intersphinx_cache_limit

Il numero massimo di giorni per cui memorizzare nella cache gli inventari remoti. Il valore di default è 5, ovvero cinque giorni. Impostarlo su un valore negativo per memorizzare nella cache gli inventari per un tempo illimitato.

intersphinx_timeout

Il numero di secondi per il timeout. Il default è `None`, ovvero senza timeout.

Nota: il timeout non è un limite di tempo per la risposta per l'intero download; piuttosto, viene sollevata un'eccezione se il server non ha emesso una risposta entro i secondi di timeout.

intersphinx_disabled_reftypes

Added in version 4.3.

Cambiato nella versione 5.0: Valore di default modificato da un elenco vuoto a `['std:doc']`.

Un elenco di stringhe può essere:

- il nome di un tipo di riferimento specifico in un dominio, ad esempio, `std:doc`, `py:func` o `cpp:class`,
- il nome di un dominio e un carattere jolly, ad esempio, `std:*`, `py:*` o `cpp:*`, or
- semplicemente un carattere jolly `*`.

Il valore di default è `['std:doc']`.

Quando un riferimento incrociato non-*external* viene risolto da intersphinx, salta la risoluzione se corrisponde a una delle specifiche in questo elenco.

Ad esempio, con `intersphinx_disabled_reftypes = ['std:doc']` un riferimento incrociato `:doc:`installation`` non tenterà di essere risolto da intersphinx, ma `:external+otherbook:doc:`installation`` si tenterà di risolvere nell'inventario denominato `otherbook` in *intersphinx_mapping*. Allo stesso tempo, per tutti i riferimenti incrociati generati, ad esempio, nelle dichiarazioni Python, si tenterà comunque di essere risolverli con intersphinx.

Se `*` è presente nell'elenco dei domini, nessun riferimento non-*external* verrà risolto da intersphinx.

Oggetti Esterni Referenziati Esplicitamente

L'estensione Intersphinx fornisce il seguente ruolo.

:external:

Added in version 4.4.

Si usa Intersphinx per eseguire la ricerca solo in progetti esterni e non nel progetto corrente. Intersphinx ha ancora bisogno di conoscere il tipo di oggetto da trovare, quindi la forma generale di questo ruolo è scrivere il riferimento incrociato come se l'oggetto fosse nel progetto corrente, ma poi anteporre il prefisso `:external`. Le due forme sono quindi

- `:external:domain:reftype:`target``, ad esempio,
`:external:py:class:`zipfile.ZipFile``, o
- `:external:reftype:`target``, ad esempio, `:external:doc:`installation``.
Con questa abbreviazione si presuppone che il dominio sia `std`.

Per limitare la ricerca a un progetto esterno specifico, viene aggiunta anche la chiave del progetto, come specificato in *intersphinx_mapping*, per ottenere le due forme

- `:external+invname:domain:reftype:`target``, per esempio,
`:external+python:py:class:`zipfile.ZipFile``, o
- `:external+invname:reftype:`target``, per esempio,
`:external+python:doc:`installation``.

Mostra tutti i link di un file di mappatura Intersphinx

Per mostrare tutti i link Intersphinx e i relativi target di un file di mappatura Intersphinx, si esegue `python -msphinx.ext.intersphinx url-or-path`. Ciò è utile quando si cerca la causa principale di un link Intersphinx interrotto in un progetto di documentazione. L'esempio seguente stampa la mappatura Intersphinx della documentazione di Python 3:

```
$ python -m sphinx.ext.intersphinx https://docs.python.org/3/objects.  
→inv
```

Utilizzo di Intersphinx con file di inventario con "Basic Authorization"

Intersphinx supporta l'autorizzazione di base in questo modo:

```
intersphinx_mapping = {'python': ('https://user:password@docs.python.  
→org/3',  
                                None)}
```

L'utente e la password verranno rimossi dall'URL durante la generazione dei link.

sphinx.ext.linkcode -- Aggiunge link esterni al codice sorgente

Autore del modulo: Pauli Virtanen

Added in version 1.2.

Questa estensione esamina le descrizioni degli oggetti (`... class::`, `... function::` ecc.) e aggiunge link esterni al codice ospitato da qualche parte sul web. L'intento è simile all'estensione `sphinx.ext.viewcode`, ma presuppone che il codice sorgente possa trovarsi da qualche parte su Internet.

Nella configurazione, si deve specificare una funzione `linkcode_resolve` che restituisce un URL in base all'oggetto.

Configurazione

linkcode_resolve

Questa è una funzione `linkcode_resolve(domain, info)`, che dovrebbe restituire l'URL al codice sorgente corrispondente all'oggetto in un dato dominio con determinate informazioni.

La funzione dovrebbe restituire `None` se non si deve aggiungere alcun link.

L'argomento `domain` specifica il dominio del linguaggio in cui si trova l'oggetto. `info` è un dizionario con la presenza garantita delle seguenti chiavi (a seconda del dominio):

- `py`: `module` (nome del modulo), `fullname` (nome dell'oggetto)
- `c`: `names` (elenco dei nomi per l'oggetto)
- `cpp`: `names` (elenco dei nomi per l'oggetto)
- `javascript`: `object` (nome dell'oggetto), `fullname` (nome dell'elemento)

Esempio:

```
def linkcode_resolve(domain, info):
    if domain != 'py':
        return None
    if not info['module']:
        return None
    filename = info['module'].replace('.', '/')
    return "https://somesite/sourcerepo/%s.py" % filename
```

Supporto matematico per output HTML in Sphinx

Added in version 0.5.

Cambiato nella versione 1.8: Il supporto matematico per i builder non HTML è integrato in sphinx-core. Quindi l'estensione mathbase non è più necessaria.

Poiché la notazione matematica non è supportata nativamente dall'HTML in alcun modo, Sphinx fornisce un supporto matematico al documento HTML con diverse estensioni. Questi utilizzano la matematica reStructuredText *directive* e *role*.

`sphinx.ext.imgmath` -- Rende le formule come immagini

Added in version 1.4.

Questa estensione esegue il rendering delle formule tramite LaTeX e `dvipng`³¹⁶ o `dvisvgm`³¹⁷ in immagini PNG o SVG. Ciò ovviamente significa che il computer su cui vengono creati i documenti deve avere entrambi i programmi disponibili.

Esistono diversi valori di configurazione impostabili per influenzare il modo in cui vengono create le immagini:

`imgmath_image_format`

Il formato dell'immagine di output. Il default è 'png'. Dovrebbe essere 'png' o 'svg'. L'immagine viene prodotta eseguendo prima `latex` sul mark-up matematico di TeX poi (a seconda del formato richiesto) `dvipng`³¹⁸ o `dvisvgm`³¹⁹.

`imgmath_use_preview`

`dvipng` e `dvisvgm` hanno entrambi la capacità di raccogliere da LaTeX la "depth" (profondità) delle formule renderizzate: un'immagine in linea dovrebbe utilizzare questa "depth" in uno stile `vertical-align` per essere correttamente allineato con il testo circostante.

Questo meccanismo richiede il `LaTeX preview package`³²⁰ (disponibile come `preview-latex-style` su Ubuntu xenial). Pertanto, il valore di default per questa opzione è `False` ma si consiglia vivamente di impostarlo su `True`.

Cambiato nella versione 2.2: Questa opzione può essere utilizzata con 'svg' `imgmath_image_format`.

`imgmath_add_tooltips`

Il default è: `True`. Se `false`, non aggiunge il codice LaTeX come un attributo "alt" per le immagini matematiche.

`imgmath_font_size`

La dimensione del font (in pt) della formula visualizzata. Il valore di default è 12. Deve essere un numero intero positivo.

³¹⁶ <https://savannah.nongnu.org/projects/dvipng/>

³¹⁷ <https://dvisvgm.de/>

³¹⁸ <https://savannah.nongnu.org/projects/dvipng/>

³¹⁹ <https://dvisvgm.de/>

³²⁰ <https://www.gnu.org/software/auctex/preview-latex.html>

imgmath_latex

Il nome del comando con cui richiamare LaTeX. Il default è 'latex'; potrebbe essere necessario impostarlo su un path completo se latex non è presente nel path di ricerca dell'eseguibile.

Dato che questa impostazione non è trasferibile da un sistema all'altro, normalmente non è utile impostarla in `conf.py`; piuttosto, dovrebbe essere preferibile fornirlo sulla riga di comando **sphinx-build** tramite l'opzione `-D`, in questo modo:

```
sphinx-build -M html -D imgmath_latex=C:\tex\latex.exe . _build
```

Questo valore dovrebbe contenere solo il path dell'eseguibile di latex, non ulteriori argomenti; utilizzare `imgmath_latex_args` per questo scopo.

Suggerimento: Per utilizzare gli [OpenType Math fonts](#)³²¹ con `unicode-math`, tramite un `imgmath_latex_preamble` personalizzato, è possibile impostare `imgmath_latex` con 'dvilualatex', ma è necessario impostare poi `imgmath_image_format` con 'svg'. Nota: questo è stato testato solo con `dvism 3.0.3`. Aumenta significativamente la durata della produzione delle immagini rispetto all'utilizzo del 'latex' standard con i tradizionali font matematici TeX.

Suggerimento: Alcuni fantasiosi markup LaTeX (è stato riportato un esempio che utilizzava TikZ per aggiungere varie decorazioni all'equazione) richiedono più esecuzioni dell'eseguibile LaTeX. Per gestire ciò, porre questa impostazione di configurazione su 'latexmk' (o un path completo ad esso) poiché questo script Perl sceglie in modo affidabile dinamicamente quante esecuzioni latex sono necessarie.

Cambiato nella versione 6.2.0: Ora è supportato l'uso di 'xelatex' (o un path completo ad esso). Ma si deve poi aggiungere '`-no-pdf`' alla lista `imgmath_latex_args` delle opzioni del comando. È richiesto 'svg' `imgmath_image_format`. Inoltre, potrebbe essere necessario che il binario `dvisvgm` sia relativamente recente (il test è stato effettuato solo con la versione 3.0.3).

Nota: Per quanto riguarda la nota precedente, attualmente non è supportato l'uso di `latexmk` con l'opzione `-xelatex`.

imgmath_latex_args

Argomenti aggiuntivi da fornire a latex, sotto forma di elenco. Il default è una lista vuota.

imgmath_latex_preamble

Codice LaTeX aggiuntivo da inserire nel preambolo dei file LaTeX utilizzati per tradurre i frammenti di formule. Questo è lasciato vuoto per default. Lo si usa per aggiungere pacchetti che modificano i font usati per le formule, come '`\usepackage{newtxsf}`' per i font sans-serif (senza grazie) o '`\usepackage{fouriernc}`' per quelli con grazie.

³²¹ <https://tex.stackexchange.com/questions/425098/which-opentype-math-fonts-are-available>

In effetti, le formule LaTeX di default hanno glifi piuttosto sottili che (nell'output HTML) spesso non corrispondono bene al font del testo.

imgmath_dvipng

Il nome del comando per invocare `dvipng`. Il default è `'dvipng'`; potrebbe essere necessario impostarlo su un path completo se `dvipng` non è nel path di ricerca degli eseguibili. Questa opzione viene utilizzata solo quando `imgmath_image_format` è impostato su `'png'`.

imgmath_dvipng_args

Argomenti aggiuntivi da fornire a `dvipng`, come elenco. Il valore di default è `['-gamma', '1.5', '-D', '110', '-bg', 'Transparent']` che crea l'immagine un po' più scura e più grande di quanto lo sia per default (questo compensa in qualche modo la sottigliezza dei font matematici LaTeX di default) e produce PNG con uno sfondo trasparente. Questa opzione viene utilizzata solo quando `imgmath_image_format` è `'png'`.

imgmath_dvisvgm

Il nome del comando per invocare `dvisvgm`. Il default è `'dvisvgm'`; potrebbe essere necessario impostarlo su un path completo se `dvisvgm` non è nel path di ricerca degli eseguibili. Questa opzione viene utilizzata solo quando `imgmath_image_format` è `'svg'`.

imgmath_dvisvgm_args

Argomenti aggiuntivi da fornire a `dvisvgm`, come elenco. Il valore predefinito è `['--no-fonts']`, il che significa che `dvisvgm` renderà i glifi come elementi del percorso (vedere le [dvisvgm FAQ](#)³²²). Questa opzione viene utilizzata solo quando `imgmath_image_format` è `'svg'`.

imgmath_embed

Il default è `False`. Se `true`, codifica le immagini di output LaTeX all'interno dei file HTML (codifica base64) e non salva file png/svg separati su disco.

Added in version 5.2.

sphinx.ext.mathjax -- Renderizza le formule tramite JavaScript

Avvertimento: La versione 4.0 cambia la versione di MathJax utilizzata nella versione 3. Potrebbe essere necessario sovrascrivere `mathjax_path` con `https://cdn.jsdelivr.net/npm/mathjax@2/MathJax.js?config=TeX-AMS-MML_HTMLorMML` o aggiornare le opzioni di configurazione per la versione 3 (vedere [mathjax3_config](#)).

Added in version 1.1.

Questa estensione inserisce la formula così com'è nei file HTML. Viene poi caricato il pacchetto JavaScript [MathJax](#)³²³ che trasforma il markup LaTeX in una formula leggibile dal vivo nel browser.

³²² <https://dvisvgm.de/FAQ>

³²³ <https://www.mathjax.org/>

Poiché MathJax (e i caratteri necessari) è molto grande, non è incluso in Sphinx ma è impostato per essere incluso automaticamente da un sito di terze parti.

Attenzione: Si dovrebbe usare la formula *directive* e *role*, non il MathJax nativo $\$, \backslash C$, ecc.

mathjax_path

Il path del file JavaScript da includere nei file HTML per caricare MathJax.

Il default è l'URL `https://` che carica i file JS dalla [jsdelivr](https://www.jsdelivr.com/)³²⁴ Content Delivery Network. Consultare la [MathJax Getting Started page](https://www.mathjax.org/#gettingstarted)³²⁵ per i dettagli. Se si vuole che MathJax sia disponibile offline o senza includere risorse da un sito di terze parti, lo si deve scaricare e impostare questo valore su un path diverso.

Il path può essere assoluto o relativo; se è relativo, è relativo alla directory `_static` dei documenti compilati.

Ad esempio, inserendo MathJax nel path statico dei documenti Sphinx, questo valore sarà `MathJax/MathJax.js`. Se si possiede più di un set di documentazione Sphinx su un server, è consigliabile installare MathJax in una posizione condivisa.

Si può anche fornire un URL `https://` diverso dall'URL del CDN.

mathjax_options

Le opzioni per scrivere il tag per mathjax. Ad esempio, si può settare l'opzione di integrità con la seguente impostazione:

```
mathjax_options = {
    'integrity': 'sha384-.....',
}
```

Il default è vuoto (`{}`).

Added in version 1.8.

Cambiato nella versione 4.4.1: Consente di modificare il metodo di caricamento (asincrono o differito) di MathJax se è impostata la chiave "async" o "defer".

mathjax3_config

Le opzioni di configurazione per MathJax v3 (utilizzato per default). Il dizionario indicato viene assegnato alla variabile JavaScript `window.MathJax`. Per ulteriori informazioni, leggere [Configuring MathJax](https://docs.mathjax.org/en/latest/web/configuration.html#configuration)³²⁶.

Il default è vuoto (non configurato).

Added in version 4.0.

³²⁴ <https://www.jsdelivr.com/>

³²⁵ <https://www.mathjax.org/#gettingstarted>

³²⁶ <https://docs.mathjax.org/en/latest/web/configuration.html#configuration>

mathjax2_config

Le opzioni di configurazione per MathJax v2 (che possono essere caricate tramite [*mathjax_path*](#)). Il valore viene utilizzato come parametro di `MathJax.Hub.Config()`. Per ulteriori informazioni, leggere [*Using in-line configuration options*](#)³²⁷.

Per esempio:

```
mathjax2_config = {
    'extensions': ['tex2jax.js'],
    'jax': ['input/TeX', 'output/HTML-CSS'],
}
```

Il default è vuoto (non configurato).

Added in version 4.0: [*mathjax_config*](#) è stato rinominato in [*mathjax2_config*](#).

mathjax_config

Nome precedente di [*mathjax2_config*](#).

Per aiutare la conversione della vecchia configurazione MathJax nella nuova [*mathjax3_config*](#), vedere [*Converting Your v2 Configuration to v3*](#)³²⁸.

Added in version 1.8.

Cambiato nella versione 4.0: Questo è stato rinominato in [*mathjax2_config*](#). [*mathjax_config*](#) è ancora supportato per compatibilità con le versioni precedenti.

sphinx.ext.jsmath -- Renderizza le formule tramite JavaScript

Questa estensione funziona esattamente come quella di MathJax, ma utilizza il vecchio pacchetto [*jsMath*](#)³²⁹. Fornisce questo valore di configurazione:

jsmath_path

Il percorso del file JavaScript da includere nei file HTML per caricare JSMath. Non c'è un default.

Il path può essere assoluto o relativo; se è relativo, è relativo alla directory `_static` dei documenti compilati.

Ad esempio, se si inserisce JSMath nel path statico dei documenti Sphinx, questo valore sarà `jsMath/easy/load.js`. Se si possiede più di un set di documentazione Sphinx su un server, è consigliabile installare jsMath in una posizione condivisa.

³²⁷ <https://docs.mathjax.org/en/v2.7-latest/configuration.html#using-in-line-configuration-options>

³²⁸ <https://docs.mathjax.org/en/latest/web/configuration.html#converting-your-v2-configuration-to-v3>

³²⁹ <https://www.math.union.edu/~dpvc/jsmath/>

sphinx.ext.napoleon -- Supporto per docstring in stile NumPy e Google

Autore del modulo: Rob Ruana

Added in version 1.3.

Panoramica

Si stanchi di scrivere docstrings come queste:

```

:param path: The path of the file to wrap
:type path: str
:param field_storage: The :class:`FileStorage` instance to wrap
:type field_storage: FileStorage
:param temporary: Whether or not to delete the file when the File
    instance is destructed
:type temporary: bool
:returns: A buffered writable file descriptor
:rtype: BufferedFileStorage

```

reStructuredText³³⁰ è fantastico, ma crea **docstrings**³³¹ visivamente dense e difficili da leggere. Confrontare il guazzabuglio sopra con la stessa cosa riscritta secondo la [Google Python Style Guide](#)³³²:

```

Args:
    path (str): The path of the file to wrap
    field_storage (FileStorage): The :class:`FileStorage` instance to
↳ wrap
    temporary (bool): Whether or not to delete the file when the File
        instance is destructed

Returns:
    BufferedFileStorage: A buffered writable file descriptor

```

Molto più leggibile, no?

Napoleon è una *extension* che consente a Sphinx di analizzare sia le docstring in stile NumPy³³³ che Google³³⁴ - lo stile consigliato dalla Khan Academy³³⁵.

Napoleon è un preprocessore che analizza le docstring in stile NumPy³³⁶ e Google³³⁷ e le converte in reStructuredText prima che Sphinx tenti di analizzarle. Ciò avviene in un passaggio inter-

³³⁰ <https://docutils.sourceforge.io/rst.html>

³³¹ <https://peps.python.org/pep-0287/>

³³² <https://google.github.io/styleguide/pyguide.html>

³³³ <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

³³⁴ <https://google.github.io/styleguide/pyguide.html#38-comments-and-docstrings>

³³⁵ <https://github.com/Khan/style-guides/blob/master/style/python.md#docstrings>

³³⁶ <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

³³⁷ <https://google.github.io/styleguide/pyguide.html#38-comments-and-docstrings>

medio mentre Sphinx elabora la documentazione, quindi non modifica nessuna delle docstring nei file del codice sorgente.

Per Iniziare

1. Dopo *setting up Sphinx* per creare i documenti, abilitare napoleon nel file Sphinx conf.py:

```
# conf.py

# Add napoleon to the extensions list
extensions = ['sphinx.ext.napoleon']
```

2. Usare sphinx-apidoc per creare la documentazione delle API:

```
$ sphinx-apidoc -f -o docs/source projectdir
```

Le Docstring

Napoleon interpreta ogni docstring che *autodoc* riesce a trovare, incluse le docstring su: modules, classes, attributes, methods, functions e variables. All'interno di ogni docstring, delle *Sections* appositamente formattate vengono analizzate e convertite in reStructuredText.

Tutta la formattazione standard di reStructuredText funziona ancora come previsto.

Le Sections Docstring

Sono supportate tutte le seguenti intestazioni di sezione:

- **Args** (*alias di Parameters*)
- **Arguments** (*alias di Parameters*)
- **Attention**
- **Attributes**
- **Caution**
- **Danger**
- **Error**
- **Example**
- **Examples**
- **Hint**
- **Important**

- Keyword Args (*alias di Keyword Arguments*)
- Keyword Arguments
- Methods
- Note
- Notes
- Other Parameters
- Parameters
- Return (*alias di Returns*)
- Returns
- Raise (*alias di Raises*)
- Raises
- References
- See Also
- Tip
- Todo
- Warning
- Warnings (*alias di Warning*)
- Warn (*alias di Warns*)
- Warns
- Yield (*alias di Yields*)
- Yields

Google contro NumPy

Napoleon supporta due stili di docstring: [Google](#)³³⁸ e [NumPy](#)³³⁹. La differenza principale tra i due stili è che Google utilizza l'indentazione per separare le sezioni, mentre NumPy utilizza la sottolineatura.

Lo stile Google:

```
def func(arg1, arg2):  
    """Summary line.  
  
    Extended description of function.
```

(continues on next page)

³³⁸ <https://google.github.io/styleguide/pyguide.html#38-comments-and-docstrings>

³³⁹ <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

(continua dalla pagina precedente)

```
Args:
    arg1 (int): Description of arg1
    arg2 (str): Description of arg2

Returns:
    bool: Description of return value

"""
return True
```

Lo stile NumPy:

```
def func(arg1, arg2):
    """Summary line.

    Extended description of function.

    Parameters
    -----
    arg1 : int
        Description of arg1
    arg2 : str
        Description of arg2

    Returns
    -----
    bool
        Description of return value

    """
    return True
```

Lo stile NumPy tende a richiedere più spazio verticale, mentre lo stile Google tende a utilizzare più spazio orizzontale. Lo stile Google tende ad essere più facile da leggere per docstring brevi e semplici, mentre lo stile NumPy tende ad essere più facile da leggere per docstring lunghe e approfondite.

La scelta tra gli stili è in gran parte estetica, ma i due stili non dovrebbero essere mescolati. Si scelga uno stile per il progetto e si resti coerenti con esso.

Vedi anche:

Per esempi completi:

- `example_google`
- `example_numpy`

Annotazione dei Tipi

PEP 484³⁴⁰ ha introdotto un modo standard per esprimere i tipi nel codice Python. Questa è un'alternativa all'espressione dei tipi direttamente nelle docstring. Un vantaggio dell'espressione dei tipi secondo **PEP 484**³⁴¹ è che i [type checker] e gli IDE possono trarne vantaggio per l'analisi statica del codice. **PEP 484**³⁴² è stato poi esteso da **PEP 526**³⁴³ che ha introdotto un modo simile di annotare le variabili (e gli attributi).

Stile Google con annotazioni di tipo in Python 3:

```
def func(arg1: int, arg2: str) -> bool:
    """Summary line.

    Extended description of function.

    Args:
        arg1: Description of arg1
        arg2: Description of arg2

    Returns:
        Description of return value

    """
    return True

class Class:
    """Summary line.

    Extended description of class

    Attributes:
        attr1: Description of attr1
        attr2: Description of attr2
    """

    attr1: int
    attr2: str
```

Stile Google con tipi in docstring:

```
def func(arg1, arg2):
    """Summary line.

    Extended description of function.
```

(continues on next page)

³⁴⁰ <https://peps.python.org/pep-0484/>

³⁴¹ <https://peps.python.org/pep-0484/>

³⁴² <https://peps.python.org/pep-0484/>

³⁴³ <https://peps.python.org/pep-0526/>

(continua dalla pagina precedente)

```
Args:
    arg1 (int): Description of arg1
    arg2 (str): Description of arg2

Returns:
    bool: Description of return value

"""
return True

class Class:
    """Summary line.

    Extended description of class

    Attributes:
        attr1 (int): Description of attr1
        attr2 (str): Description of attr2
    """
```

Nota: Le [Python 2/3 compatible annotations](#)³⁴⁴ non sono attualmente supportate da Sphinx e non verranno visualizzate nella documentazione.

Configurazione

Di seguito sono elencate tutte le impostazioni utilizzate da Napoleon e i relativi default. Queste impostazioni sono modificabili nel file Sphinx `conf.py`. Assicurarsi che "sphinx.ext.napoleon" sia abilitato in `conf.py`:

```
# conf.py

# Add any Sphinx extension module names here, as strings
extensions = ['sphinx.ext.napoleon']

# Napoleon settings
napoleon_google_docstring = True
napoleon_numpy_docstring = True
napoleon_include_init_with_doc = False
napoleon_include_private_with_doc = False
napoleon_include_special_with_doc = True
napoleon_use_admonition_for_examples = False
```

(continues on next page)

³⁴⁴ <https://peps.python.org/pep-0484/#suggested-syntax-for-python-2-7-and-straddling-code>

(continua dalla pagina precedente)

```

napoleon_use_admonition_for_notes = False
napoleon_use_admonition_for_references = False
napoleon_use_ivar = False
napoleon_use_param = True
napoleon_use_rtype = True
napoleon_preprocess_types = False
napoleon_type_aliases = None
napoleon_attr_annotations = True

```

napoleon_google_docstring

True per analizzare le docstring in [stile Google](#)³⁴⁵. False per disabilitare il supporto per le docstring in stile Google. *Il default è True.*

napoleon_numpy_docstring

True per analizzare le docstring in [stile NumPy](#)³⁴⁶. False per disabilitare il supporto per le docstring in stile NumPy. *Il default è True.*

napoleon_include_init_with_doc

True per elencare le docstring `__init__` separatamente dalla classe docstring. False per ricorrere al comportamento di default di Sphinx, che considera la docstring `__init__` come parte della documentazione della classe. *Il default è False.*

Se True:

```

def __init__(self):
    """
    This will be included in the docs because it has a docstring
    """

def __init__(self):
    # This will NOT be included in the docs

```

napoleon_include_private_with_doc

True per includere membri privati (come `_membername`) con le docstring nella documentazione. Falso per ricorrere al comportamento di default di Sphinx. *Il default è False.*

Se True:

```

def _included(self):
    """
    This will be included in the docs because it has a docstring
    """
    pass

```

(continues on next page)

³⁴⁵ <https://google.github.io/styleguide/pyguide.html#38-comments-and-docstrings>

³⁴⁶ <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

(continua dalla pagina precedente)

```
def _skipped(self):
    # This will NOT be included in the docs
    pass
```

napoleon_include_special_with_doc

True per includere membri speciali (come `__membername__`) con le docstring nella documentazione. Falso per ricorrere al comportamento di default di Sphinx. *Il default è True.*

Se True:

```
def __str__(self):
    """
    This will be included in the docs because it has a docstring
    """
    return unicode(self).encode('utf-8')

def __unicode__(self):
    # This will NOT be included in the docs
    return unicode(self.__class__.__name__)
```

napoleon_use_admonition_for_examples

True per usare `.. admonition::` come direttiva per le sezioni **Example** ed **Examples**. False per usare `.. rubric::`. Uno potrebbe sembrare migliore dell'altro a seconda del tema HTML utilizzato. *Il default è False.*

Questo snippet in stile NumPy³⁴⁷ verrà convertito come segue:

```
Example
-----
This is just a quick example
```

Se True:

```
.. admonition:: Example

    This is just a quick example
```

Se False:

```
.. rubric:: Example

This is just a quick example
```

napoleon_use_admonition_for_notes

True per usare `.. admonition::` come direttiva per le sezioni **Notes**. False per usare `.. rubric::`. *Il default è False.*

³⁴⁷ <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

Nota: La sezione singolare **Note** verrà sempre convertita in `.. note::` come direttiva.

Vedi anche:

[napoleon_use_admonition_for_examples](#)

napoleon_use_admonition_for_references

True per usare `.. admonition::` come direttiva per le sezioni **References**. False per usare `.. rubric::`. *Il default è False.*

Vedi anche:

[napoleon_use_admonition_for_examples](#)

napoleon_use_ivar

True per usare il ruolo `:ivar:` per le variabili di istanza. False per usare `.. attribute::` come direttiva. *Il default è False.*

Questo snippet in stile NumPy³⁴⁸ verrà convertito come segue:

```
Attributes
-----
attr1 : int
    Description of `attr1`
```

Se True:

```
:ivar attr1: Description of `attr1`
:vartype attr1: int
```

Se False:

```
.. attribute:: attr1

    Description of `attr1`

    :type: int
```

napoleon_use_param

True per usare un ruolo `:param:` per ciascun parametro di funzione. False per utilizzare un singolo ruolo `:parameters:` per tutti i parametri. *Il default è True.*

Questo snippet in stile NumPy³⁴⁹ verrà convertito come segue:

```
Parameters
-----
arg1 : str
    Description of `arg1`
```

(continues on next page)

³⁴⁸ <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

(continua dalla pagina precedente)

```
arg2 : int, optional
    Description of `arg2`, defaults to 0
```

Se True:

```
:param arg1: Description of `arg1`
:type arg1: str
:param arg2: Description of `arg2`, defaults to 0
:type arg2: :class:`int`, *optional*
```

Se False:

```
:parameters: * **arg1** (*str*) --
    Description of `arg1`
    * **arg2** (*int, optional*) --
    Description of `arg2`, defaults to 0
```

napoleon_use_keyword

True per usare un ruolo `:keyword:` per ogni argomento di parola chiave della funzione. False per usare un unico ruolo `:keyword arguments:` per tutte le parole chiave. *Il default è True.*

Questo si comporta in modo simile a [napoleon_use_param](#). Notare che a differenza dei docutil, `:keyword:` e `:param:` non verranno trattati allo stesso modo: ci sarà una sezione "Keyword Arguments" separata, resa allo stesso modo della sezione "Parameters" (se possibile, creati come i link dei tipi)

Vedi anche:[napoleon_use_param](#)**napoleon_use_rtype**

True per usare il ruolo `:rtype:` per il tipo restituito. False l'output del tipo restituito in linea con la descrizione. *Il default è True.*

Questo snippet in stile NumPy³⁴⁹ verrà convertito come segue:

```
Returns
-----
bool
    True if successful, False otherwise
```

Se True:

```
:returns: True if successful, False otherwise
:rtype: bool
```

Se False:

³⁴⁹ <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

```
:returns: *bool* -- True if successful, False otherwise
```

napoleon_preprocess_types

True per convertire le definizioni di tipo nelle docstring come riferimenti. Il default è *False*.

Added in version 3.2.1.

Cambiato nella versione 3.5: Preelabora anche le docstring in stile Google.

napoleon_type_aliases

Una mappatura per tradurre i nomi dei tipi in altri nomi o riferimenti. Funziona solo quando `napoleon_use_param = True`. *Il default è None*.

Con:

```
napoleon_type_aliases = {
    "CustomType": "mypackage.CustomType",
    "dict-like": ":term:`dict-like <mapping>`",
}
```

Questo snippet [NumPy style](#)³⁵¹:

```
Parameters
-----
arg1 : CustomType
    Description of `arg1`
arg2 : dict-like
    Description of `arg2`
```

diventa:

```
:param arg1: Description of `arg1`
:type arg1: mypackage.CustomType
:param arg2: Description of `arg2`
:type arg2: :term:`dict-like <mapping>`
```

Added in version 3.2.

napoleon_attr_annotations

True per consentire l'uso delle annotazioni degli attributi [PEP 526](#)³⁵² nelle classi. Se un attributo è documentato nella docstring senza un tipo e ha un'annotazione nel corpo della classe, viene utilizzato quel tipo.

Added in version 3.4.

³⁵⁰ <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

³⁵¹ <https://numpydoc.readthedocs.io/en/latest/format.html#docstring-standard>

³⁵² <https://peps.python.org/pep-0526/>

`napoleon_custom_sections`

Aggiunge un elenco di sezioni personalizzate da includere, espandendo l'elenco delle sezioni analizzate. *Il default è None.*

Le voci possono essere stringhe o tuple, a seconda dell'intenzione:

- Per creare una sezione "generic" personalizzata è sufficiente passare una stringa.
- Per creare un alias per una sezione esistente, si passa una tupla contenente il nome dell'alias e l'originale, in quest'ordine.
- Per creare una sezione personalizzata che venga visualizzata come la sezione dei parametri o quella dei 'return', si passa una tupla contenente il nome della sezione personalizzata e un valore stringa, "params_style" o "returns_style".

Se una voce è solo una stringa, viene interpretata come un'intestazione per una sezione generica. Se la voce è una tupla/lista/contenitore indicizzato, la prima voce è il nome della sezione, la seconda è la chiave della sezione da emulare. Se il secondo valore della voce è "params_style" o "returns_style", la sezione personalizzata verrà visualizzata come la sezione parametri o la sezione dei return.

Added in version 1.8.

Cambiato nella versione 3.5: Supporta `params_style` e `returns_style`

`sphinx.ext.todo` -- Supporto per gli elementi todo

Autore del modulo: Daniel Bültmann

Added in version 0.5.

Ci sono due direttive aggiuntive quando si utilizza questa estensione:

`.. todo::`

Utilizzare questa direttiva come, ad esempio, *note*.

Verrà visualizzato nell'output solo se `todo_include_todos` è True.

Added in version 1.3.2: Questa direttiva supporta un'opzione `class` che determina l'attributo `class` per l'output HTML. Se non specificato, la classe viene impostata per default su `admonition-todo`.

`.. todolist::`

Questa direttiva è sostituita da un elenco di tutte le direttive todo nell'intera documentazione, se `todo_include_todos` è True.

Questi possono essere configurati come mostrato di seguito.

Configurazione

`todo_include_todos`

Se è True, `todo` e `todolist` producono un output, altrimenti non producono nulla. Il default è False.

`todo_emit_warnings`

Se è True, `todo` emettono un warning per ogni voce TODO. Il default è False.

Added in version 1.5.

`todo_link_only`

Se è True, `todolist` produce un output senza il path e la riga del file, il valore di default è False.

Added in version 1.4.

autodoc fornisce il seguente evento aggiuntivo:

`todo-defined(app, node)`

Added in version 1.5.

Emesso quando viene definito un todo. `node` è il nodo `sphinx.ext.todo.todo_node` definito.

`sphinx.ext.viewcode` -- Aggiunge i link al codice sorgente evidenziato

Autore del modulo: Georg Brandl

Added in version 1.0.

Questa estensione esamina le descrizioni degli oggetti Python (`.. class::`, `.. function::` ecc.) e tenta di trovare i file sorgente in cui sono contenuti gli oggetti. Una volta trovato, verrà generata una pagina HTML separata per ciascun modulo con una versione evidenziata del codice sorgente e verrà aggiunto un link a tutte le descrizioni degli oggetti che porta al codice sorgente dell'oggetto descritto. Verrà inoltre inserito un link dal sorgente alla descrizione.

Avvertimento: Fondamentalmente, l'estensione `viewcode` importerà i moduli a cui è collegata. Se qualche modulo ha effetti collaterali durante l'importazione, questi verranno eseguiti con l'esecuzione di `sphinx-build`.

Se si documentano gli script (al contrario dei moduli della libreria), la loro routine principale dev'essere protetta da una condizione `if __name__ == '__main__':`.

Inoltre, se per non importare i moduli tramite `viewcode`, si può indicare la posizione del codice sorgente a `viewcode` con l'evento `viewcode-find-source`.

Se `viewcode_follow_imported_members` è abilitato, si devono anche risolvere gli attributi importati utilizzando l'evento `viewcode-follow-imported`.

Questa estensione funziona solo su builder correlati all'HTML come `html`, `applehelp`, `devhelp`, `htmlhelp`, `qthelp` e così via tranne `singlehtml`. Per default il builder `epub` non supporta questa estensione (vedere [viewcode_enable_epub](#)).

Configurazione

`viewcode_follow_imported_members`

Se è `True`, l'estensione `viewcode` emetterà l'evento `viewcode-follow-imported` per risolvere il nome del modulo da altre estensioni.. Il default è `True`.

Added in version 1.3.

Cambiato nella versione 1.8: Rinominato da `viewcode_import` a `viewcode_follow_imported_members`.

`viewcode_enable_epub`

Se è `True`, anche l'estensione `viewcode` è abilitata anche se si usano i builder `epub`. Questa estensione genera pagine esterne a `toctree`, ma non è preferibile come formato `epub`.

Fino alla versione 1.4.x, questa estensione è sempre abilitata. Per generare un `epub` uguale a 1.4.x, si deve impostare a `"True"`, ma il punteggio del controllo del formato `epub` peggiora.

Il default è `False`.

Added in version 1.5.

Avvertimento: Non tutti i lettori `epub` supportano le pagine generate dall'estensione `viewcode`. Tali lettori ignorano i link a pagine che non sono sotto `toctree`.

I risultati del rendering di alcuni lettori sono danneggiati e il punteggio di `epubcheck`³⁵³ peggiora anche se il lettore supporta.

`viewcode_line_numbers`

Il default è `False`.

Se impostato su `True`, i numeri di riga in linea verranno aggiunti al codice evidenziato.

Added in version 7.2.

`viewcode-find-source`(*app*, *modname*)

Added in version 1.8.

Trova il codice sorgente per un modulo. Un gestore eventi per questo evento dovrebbe restituire una tupla del codice sorgente stesso e un dizionario di tag. Il dizionario mappa il nome di una classe, funzione, attributo, ecc. su una tupla del suo tipo, il numero della riga iniziale e il numero della riga finale. Il tipo dovrebbe essere `"class"`, `"def"` o `"other"`.

Parametri

³⁵³ <https://github.com/IDPF/epubcheck>

- **app** -- L'oggetto dell'applicazione Sphinx.
- **modname** -- Il nome del modulo per cui trovare il codice sorgente.

viewcode-follow-imported(*app, modname, attribute*)

Added in version 1.8.

Trova il nome del modulo originale per un attributo.

Parametri

- **app** -- L'oggetto dell'applicazione Sphinx.
- **modname** -- Il nome del modulo a cui appartiene l'attributo.
- **attribute** -- Il nome del membro da seguire.

Estensioni di terze parti

Si possono trovare diverse estensioni fornite dagli utenti nell'organizzazione [sphinx-contrib](https://github.com/sphinx-contrib)³⁵⁴. Per includere la propria estensione in questa organizzazione, seguire semplicemente le istruzioni fornite nel progetto [github-administration](https://github.com/sphinx-contrib/github-administration)³⁵⁵. Questo è facoltativo e ci sono diverse estensioni ospitate altrove. I progetti [awesome-sphinxdoc](https://github.com/yoloseem/awesome-sphinxdoc)³⁵⁶ e [sphinx-extensions](https://sphinx-extensions.readthedocs.io/en/latest/)³⁵⁷ sono entrambi elenchi curati di pacchetti Sphinx e molti pacchetti utilizzano i classificatori di ricerca `Framework :: Sphinx :: Extension`³⁵⁸ e `Framework :: Sphinx :: Theme`³⁵⁹ per le estensioni e i temi Sphinx, rispettivamente.

Dove mettere le proprie estensioni?

Le estensioni locali ad un progetto dovrebbero essere inserite nella struttura di directory del progetto. Imposta il path di ricerca dei moduli Python, `sys.path`, di conseguenza in modo che Sphinx possa trovarli. Ad esempio, se l'estensione `foo.py` si trova nella sottodirectory `exts` della root del progetto, lo si mette in `conf.py`:

```
import sys, os

sys.path.append(os.path.abspath('exts'))

extensions = ['foo']
```

Si possono anche installare estensioni ovunque su `sys.path`, ad es. nella directory `site-packages`.

³⁵⁴ <https://github.com/sphinx-contrib/>

³⁵⁵ <https://github.com/sphinx-contrib/github-administration>

³⁵⁶ <https://github.com/yoloseem/awesome-sphinxdoc>

³⁵⁷ <https://sphinx-extensions.readthedocs.io/en/latest/>

³⁵⁸ <https://pypi.org/search/?c=Framework+%3A+Sphinx+%3A+Extension>

³⁵⁹ <https://pypi.org/search/?c=Framework+%3A+Sphinx+%3A+Theme>

4.8 Temi HTML

Sphinx fornisce una serie di builder per formati HTML e basati su HTML.

I Builder

Da fare: Popolato quando il documento 'builders' viene suddiviso.

Temi

Added in version 0.6.

Nota: Questa sezione fornisce informazioni sull'utilizzo di temi HTML preesistenti. Per creare il proprio tema, fare riferimento a *Sviluppo di temi HTML*.

Sphinx supporta la modifica dell'aspetto del suo output HTML tramite i *temi*. Un tema è una raccolta di template HTML, fogli di stile e altri file statici. Inoltre, ha un file di configurazione che specifica da quale tema ereditare, quale stile di evidenziazione utilizzare e quali opzioni esistono per personalizzare l'aspetto del tema.

I temi sono pensati per essere inconsapevoli del progetto, quindi possono essere utilizzati per progetti diversi senza modifiche.

Usare un tema

Usare un *tema fornito con Sphinx* è semplice. Dato che non è necessario installarli, se ne deve solo impostare il valore di configurazione `html_theme`. Ad esempio, per abilitare il tema `classic`, si aggiunge quanto segue a `conf.py`:

```
html_theme = "classic"
```

Si possono anche impostare opzioni specifiche del tema utilizzando il valore di configurazione `html_theme_options`. Queste opzioni vengono generalmente utilizzate per modificare l'aspetto del tema. Ad esempio, per posizionare la barra laterale sul lato destro e uno sfondo nero per la barra delle relazioni (la barra con i link di navigazione nella parte superiore e inferiore della pagina), si aggiunge quanto segue in `conf.py`:

```
html_theme_options = {
    "rightsidebar": "true",
    "relbarbgcolor": "black"
}
```

Se il tema non viene fornito con Sphinx, può essere in due [form] statiche o come pacchetto Python. Per i moduli statici, è supportata una directory (contenente `theme.toml` e altri file necessari) o un file zip con lo stesso contenuto. La directory o il file zip deve essere inserito dove Sphinx può trovarlo; per questo c'è il valore di configurazione `html_theme_path`. Può essere un elenco di directory, relative alla directory contenente `conf.py`, che può contenere directory di temi o file zip. Ad esempio, se si ha un tema nel file `blue.zip`, lo si può inserire direttamente nella directory contenente `conf.py` e utilizzare questa configurazione:

```
html_theme = "blue"
html_theme_path = ["."]
```

La terza forma è un pacchetto Python. Se un tema che si vuol utilizzare è distribuito come pacchetto Python, lo si può usare dopo l'installazione

```
# installing theme package
$ pip install sphinxjp.themes.dotted
```

Una volta installato, può essere utilizzato allo stesso modo di una directory o di un tema basato su file zip:

```
html_theme = "dotted"
```



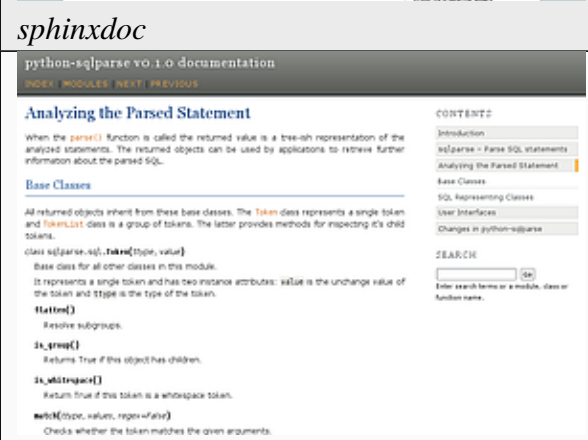

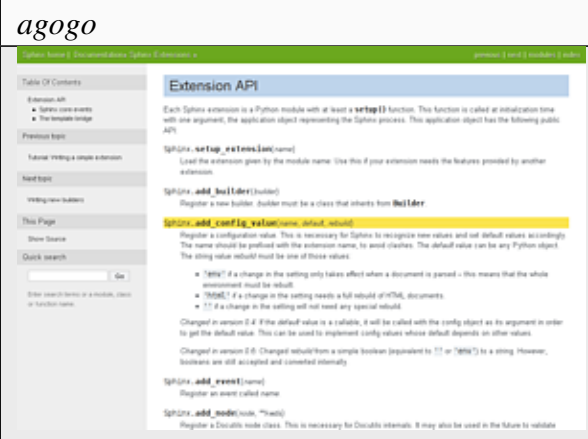



Per ulteriori informazioni sulla progettazione dei temi, incluse le informazioni su come scrivere i propri temi, fare riferimento a *Sviluppo di temi HTML*.

Temi nativi

Panoramica dei temi	
<p><i>alabaster</i></p>	<p><i>classic</i></p>

continues on next page

Tabella 1 – continua dalla pagina precedente

	
<p><i>sphinxdoc</i></p> 	<p><i>scrolls</i></p> 
<p><i>agogo</i></p> 	<p><i>traditional</i></p> 
<p><i>nature</i></p> 	<p><i>haiku</i></p> 
<p><i>pyramid</i></p>	<p><i>bizstyle</i></p>

Sphinx viene fornito con una selezione di temi tra cui scegliere.

Notare che di questi temi solo Alabaster e Scrolls sono ottimizzati per dispositivi mobili, gli altri temi ricorrono allo scorrimento orizzontale se lo schermo è troppo stretto.

Questi temi sono:

basic

Questo è un layout basilare senza stile utilizzato come base per gli altri temi e utilizzabile anche come base per temi personalizzati. L'HTML contiene tutti gli elementi importanti come la barra laterale e la barra delle relazioni. Ci sono queste opzioni (che vengono ereditate dagli altri temi):

- **nosidebar** (true o false): Non includere la barra laterale. Il default è False.
- **sidebarwidth** (int o str): Larghezza della barra laterale in pixel. Può essere un int, che viene interpretato come pixel o una stringa di dimensione CSS valida come '70em' o '50%'. Il default è 230 pixel.
- **body_min_width** (int o str): Larghezza minima del corpo del documento. Può essere un int, che viene interpretato come pixel o una stringa di dimensione CSS valida come '70em' o '50%'. Usare 0 se non si vuole limitare la larghezza. Il default può dipendere dal tema (spesso 450px).
- **body_max_width** (int o str): Larghezza massima del corpo del documento. Può essere un int, che viene interpretato come pixel o una stringa di dimensione CSS valida come '70em' o '50%'. Usare 'none' per non limitare la larghezza. Il default può dipendere dal tema (spesso 800px).
- **navigation_with_keys** (true o false): Consente la navigazione con le seguenti scorciatoie da tastiera:
 - Left arrow: pagina precedente
 - Right arrow: pagina successiva

Il default è False.

- **enable_search_shortcuts** (true o false): Consente di passare alla casella di ricerca con / e di rimuovere l'evidenziazione della ricerca con Esc.

Il default è True.

- **globaltoc_collapse** (true o false): Espande solo le sottosezioni del documento corrente in `globaltoc.html` (vedere [html_sidebars](#)). Il default è True.

Added in version 3.1.

- **globaltoc_includehidden** (true o false): Mostra anche quelle sottosezioni in `globaltoc.html` (vedere [html_sidebars](#)) che sono state incluse con il flag `:hidden:` della direttiva `toctree`. Il default è False.

Added in version 3.1.

- **globaltoc_maxdepth** (int): La profondità massima del `toctree` in `globaltoc.html` (vedere [html_sidebars](#)). Settarlo a -1 per consentire una profondità illimitata. Il default è la profondità massima selezionata nella direttiva `toctree`.

Added in version 3.2.

alabaster

Il tema [Alabaster](https://pypi.org/project/alabaster/)³⁶⁰ è un tema "Kr" di Sphinx modificato da @kennethreitz (specialmente come usato nel suo progetto Requests), a sua volta originariamente basato sul tema di @mitsuhiko, utilizzato per Flask e progetti correlati. Fare riferimento alla sua [installation page](https://alabaster.readthedocs.io/en/latest/installation.html)³⁶¹ per informazioni su come configurare `html_sidebars` per il suo utilizzo.

classic

Questo è il tema classico, che assomiglia alla [documentazione di Python 2](https://docs.python.org/2/)³⁶². È personalizzabile tramite queste opzioni:

- **rightsidebar** (true o false): Posiziona la barra laterale sul lato destro. Il default è `False`.
- **stickysidebar** (true o false): Rende la barra laterale "fissa" in modo che non scorra fuori dalla schermata per un corpo con tanti contenuti. Potrebbe non funzionare bene con tutti i browser. Il default è `False`.
- **collapsiblesidebar** (true o false): Aggiunge uno snippet JavaScript *sperimentale* che renda la barra laterale comprimibile tramite un pulsante laterale. Il default è `False`.
- **externalrefs** (true o false): Visualizza i link esterni diversamente da quelli interni. Il default è `False`.

Esistono anche varie opzioni di colore e font che possono modificare la combinazione di colori senza dover scrivere un foglio stile personalizzato:

- **footerbgcolor** (CSS color): Colore di sfondo per la riga del piè di pagina.
- **footertextcolor** (CSS color): Colore del testo per la riga del piè di pagina.
- **sidebarbgcolor** (CSS color): Colore di sfondo per la barra laterale.
- **sidebarbtncolor** (CSS color): Colore di sfondo per il pulsante di compressione della barra laterale (utilizzato quando `collapsiblesidebar` è `True`).
- **sidebartextcolor** (CSS color): Colore del testo per la barra laterale.
- **sidebarlinkcolor** (CSS color): Colore del link per la barra laterale.
- **relbarbgcolor** (CSS color): Colore di sfondo per la barra delle relazioni.
- **relbartextcolor** (CSS color): Colore del testo per la barra delle relazioni.
- **relbarlinkcolor** (CSS color): Colore del link per la barra delle relazioni.
- **bgcolor** (CSS color): Colore di sfondo del body.
- **textcolor** (CSS color): Colore del testo nel body.
- **linkcolor** (CSS color): Colore del link nel body.
- **visitedlinkcolor** (CSS color): Colore dei link visitati nel body.
- **headbgcolor** (CSS color): Colore di sfondo per i titoli.

³⁶⁰ <https://pypi.org/project/alabaster/>

³⁶¹ <https://alabaster.readthedocs.io/en/latest/installation.html>

³⁶² <https://docs.python.org/2/>

- **headtextcolor** (CSS color): Colore del testo per i titoli.
- **headlinkcolor** (CSS color): Colore del link per i titoli.
- **codebgcolor** (CSS color): Colore di sfondo per i blocchi di codice.
- **codetextcolor** (CSS color): Colore di default del testo per i blocchi di codice, se non impostato diversamente dallo stile di evidenziazione.
- **bodyfont** (CSS font-family): Font per testo normale.
- **headfont** (CSS font-family): Font per i titoli.

sphinxdoc

Il tema originariamente utilizzato da questa documentazione. Ha una barra laterale sul lato destro. Al momento non ci sono opzioni oltre a *nosidebar* e *sidebarwidth*.

Nota: La documentazione di Sphinx ora utilizza [una versione modificata del tema sphinxdoc](#)³⁶³.

scrolls

Un tema più leggero, basato sulla [documentazione Jinja](#)³⁶⁴. Sono disponibili le seguenti opzioni di colore:

- **headerbordercolor**
- **subheadlinecolor**
- **linkcolor**
- **visitedlinkcolor**
- **admonitioncolor**

agogo

Un tema creato da Andi Albrecht. Sono supportate le seguenti opzioni:

- **bodyfont** (CSS font family): Font per il testo normale..
- **headerfont** (CSS font family): Font per i titoli.
- **pagewidth** (CSS length): Larghezza del contenuto della pagina, il default è 70em.
- **documentwidth** (CSS length): Larghezza del documento (senza barra laterale), il default è 50em.
- **sidebarwidth** (CSS length): Larghezza della barra laterale, il default 20em.
- **rightsidebar** (true o false): Posiziona la barra laterale sul lato destro. Il default è True.
- **bgcolor** (CSS color): Colore dello sfondo.
- **headerbg** (CSS value for "background"): Sfondo per l'area dell'intestazione, per default una sfumatura grigiastra.

³⁶³ https://github.com/sphinx-doc/sphinx/tree/master/doc/_themes/sphinx13

³⁶⁴ <https://jinja.palletsprojects.com/>

- **footerbg** (CSS value for "background"): Sfondo per l'area del piè di pagina, per default un gradiente grigio chiaro.
- **linkcolor** (CSS color): Colore del link nel body.
- **headercolor1**, **headercolor2** (CSS color): Colori per titoli <h1> e <h2>.
- **headerlinkcolor** (CSS color): Colore per il link di riferimento nei titoli.
- **textalign** (CSS *text-align* value): Allineamento del testo nel body, il default è justify.

nature

Un tema verdastro. Al momento non ci sono opzioni oltre a *nosidebar* e *sidebarwidth*.

pyramid

Un tema del progetto del framework web Pyramid, progettato da Blaise Laflamme. Al momento non ci sono opzioni oltre a *nosidebar* e *sidebarwidth*.

haiku

Un tema senza barra laterale ispirato alla [guida utente di Haiku OS](https://www.haiku-os.org/docs/userguide/en/contents.html)³⁶⁵. Sono supportate le seguenti opzioni:

- **full_logo** (true o false, il default è False): Se è true, l'intestazione mostrerà solo il [html_logo](#). Lo si usa per loghi di grandi dimensioni. Se questo è false, il logo (se presente) verrà mostrato mobile a destra e il titolo della documentazione verrà inserito nell'intestazione.
- **textcolor**, **headingcolor**, **linkcolor**, **visitedlinkcolor**, **hoverlinkcolor** (CSS colors): Colori per vari elementi del body.

traditional

Un tema che ricorda la vecchia documentazione Python. Al momento non ci sono opzioni oltre a *nosidebar* e *sidebarwidth*.

epub

Un tema per il builder epub. Questo tema cerca di risparmiare spazio visivo che è una risorsa scarsa sui lettori di ebook. Sono supportate le seguenti opzioni:

- **relbar1** (true o false, per default True): Se è true, il blocco `relbar1` viene inserito nell'output di epub, altrimenti viene omissso.
- **footer** (true o false, per default True): Se è true, il blocco `footer` viene inserito nell'output di epub, altrimenti viene omissso.

bizstyle

Un semplice tema bluastro. Le seguenti opzioni sono supportate oltre a *nosidebar* e a *sidebarwidth*:

- **rightsidebar** (true o false): Posiziona la barra laterale sul lato destro. Il default è False.

Added in version 1.3: 'alabaster', 'sphinx_rtd_theme' e 'bizstyle' theme.

³⁶⁵ <https://www.haiku-os.org/docs/userguide/en/contents.html>

Cambiato nella versione 1.3: Il tema di 'default' è stato rinominato 'classic'. 'default' è ancora disponibile, tuttavia verrà emesso un avviso che si tratta di un alias per il nuovo tema 'alabaster'.

Temi di Terze Parti

Esistono molti temi di terze parti creati per Sphinx. Alcuni di questi sono per uso generale, mentre altri sono specifici per un singolo progetto.

sphinx-themes.org³⁶⁶ è una galleria che mostra vari temi per Sphinx, con documentazione demo presentata sotto ciascun tema. I temi si trovano anche su [PyPI](https://pypi.org/search/?q=&o=&c=Framework+%3A+Sphinx+%3A+Theme)³⁶⁷ (usando il classificatore `Framework :: Sphinx :: Theme`), [GitHub](https://github.com/search?utf8=%E2%9C%93&q=sphinx+theme)³⁶⁸ e [GitLab](https://gitlab.com/explore?name=sphinx+theme)³⁶⁹.

4.9 Internazionalizzazione

Added in version 1.1.

Oltre alle traduzioni fornite per i messaggi generati da Sphinx come le barre di navigazione, Sphinx fornisce meccanismi che facilitano la traduzione di *documenti*. Vedere le *Opzioni per l'internazionalizzazione* per i dettagli sulla configurazione.

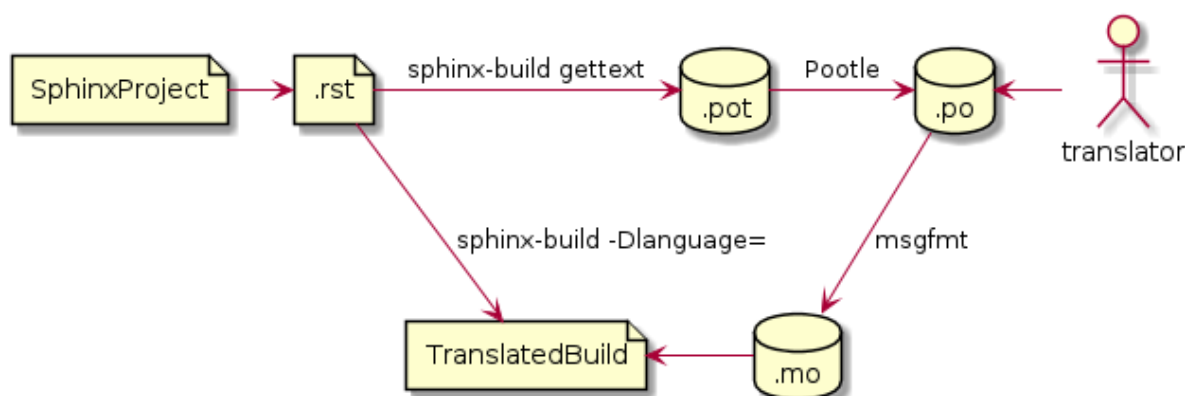


Fig. 1: Visualizzazione del flusso di lavoro delle traduzioni in Sphinx. (La figura è stata creata da [plantuml](https://plantuml.com)³⁷⁰).

- *Dettagli sull'internazionalizzazione di Sphinx*
- *Tradurre con sphinx-intl*
 - *Guida rapida*
 - *Traduzione*

³⁶⁶ <https://sphinx-themes.org/>

³⁶⁷ <https://pypi.org/search/?q=&o=&c=Framework+%3A+Sphinx+%3A+Theme>

³⁶⁸ <https://github.com/search?utf8=%E2%9C%93&q=sphinx+theme>

³⁶⁹ <https://gitlab.com/explore?name=sphinx+theme>

³⁷⁰ <https://plantuml.com>

– Aggiornare i file po con i nuovi file pot

- Utilizzo del servizio Transifex per team di traduzione
- Utilizzo del servizio Transifex per team di traduzione
- Contribuire alla traduzione dei reference di Sphinx
- Avanzamento e statistiche della traduzione

Dettagli sull'internazionalizzazione di Sphinx

gettext¹ è uno standard consolidato per l'internazionalizzazione e la localizzazione. Mappa semplicemente i messaggi in un programma su una stringa tradotta. Sphinx usa queste funzionalità per tradurre interi documenti.

Inizialmente i manutentori del progetto devono raccogliere tutte le stringhe traducibili (chiamate anche *messaggi*) per fornirle ai traduttori. Sphinx li estrae invocando **sphinx-build -M gettext**.

Ogni singolo elemento nel doctree finirà in un singolo messaggio, il che si tradurrà in elenchi equamente divisi in blocchi diversi mentre i paragrafi di grandi dimensioni rimarranno grossolani come lo erano nel documento originale. Ciò garantisce aggiornamenti fluidi dei documenti fornendo allo stesso tempo un po' di contesto ai traduttori nei passaggi di testo libero. È compito del manutentore suddividere i paragrafi troppo grandi poiché non esiste un modo automatizzato per farlo.

Dopo che Sphinx ha eseguito con successo *MessageCatalogBuilder* si troverà una raccolta di file .pot nella directory di output. Questi sono **catalog templates** e contengono messaggi nella *sola* lingua originale.

Possono essere consegnati ai traduttori che li trasformeranno in file .po --- i cosiddetti **message catalogs** --- contenenti una mappatura dai messaggi originali a stringhe in lingua straniera.

gettext li compila in un formato binario noto come **binary catalogs** tramite **msgfmt** per ragioni di efficienza. Se questi file vengono resi rilevabili con *locale_dirs* per il proprio *language*, Sphinx li raccoglierà automaticamente.

Un esempio: si ha un documento `usage.rst` nel progetto Sphinx. Il builder *gettext* inserirà i suoi messaggi in `usage.pot`. Si immagini di avere traduzioni in spagnolo² archiviate in `usage.po` --- affinché le build vengano tradotte si devono seguire queste istruzioni:

- Compilare il catalogo dei messaggi in una directory locale, ad esempio `locale`, in modo che finisca in `./locale/es/LC_MESSAGES/usage.mo` nella directory di origine (dove `es` è il codice per lo spagnolo).

```
msgfmt "usage.po" -o "locale/es/LC_MESSAGES/usage.mo"
```

- Si imposta *locale_dirs* su `["locale/"]`.

¹ Per i dettagli su quella suite software, vedere [GNU gettext utilities](#)³⁸³.

³⁸³ <https://www.gnu.org/software/gettext/manual/gettext.html#Introduction>

² Perché nessuno si aspetta l'Inquisizione Spagnola!

- Si imposta `language` su `es` (è possibile anche tramite `-D`).
- Si esegue la build desiderata.

Per proteggersi dagli errori, viene emesso un warning se i riferimenti incrociati nel paragrafo tradotto non corrispondono a quelli dell'originale. Questo può essere disattivato globalmente utilizzando la variabile di configurazione `suppress_warnings`. In alternativa, per disattivarlo solo per un messaggio, si termina il messaggio con `#noqa` in questo modo:

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
risus tortor, luctus id ultrices at. #noqa

```

(Scrivere `\#noqa` nel caso in cui si voglia avere `"#noqa"` letteralmente nel testo. Questo non si applica ai blocchi di codice, dove `#noqa` viene ignorato perché i blocchi di codice non contengono comunque riferimenti).

Added in version 4.5: Il meccanismo `#noqa`.

Tradurre con sphinx-intl

Guida rapida

`sphinx-intl`³⁷¹ è uno strumento utile per lavorare con il flusso di traduzione di Sphinx. Questa sezione descrive un modo semplice per tradurre con `sphinx-intl`.

1. Installare `sphinx-intl`³⁷².

```
$ pip install sphinx-intl
```

2. Aggiungere le configurazioni a `conf.py`.

```

locale_dirs = ['locale/']    # path is example but recommended.
gettext_compact = False      # optional.

```

Questo caso di studio presuppone che `BUILDDIR` sia impostato a `_build`, `locale_dirs` sia impostato a `locale/` e `gettext_compact` sia impostato a `False` (il documento Sphinx è già configurato come tale).

3. Estrazione dei messaggi traducibili in file pot.

```
$ make gettext
```

I file pot generati verranno inseriti nella directory `_build/gettext`.

4. Generazione dei file po.

Utilizzeremo i file pot generati nel passaggio precedente.

³⁷¹ <https://pypi.org/project/sphinx-intl/>

³⁷² <https://pypi.org/project/sphinx-intl/>

```
$ sphinx-intl update -p _build/gettext -l de -l ja
```

Una volta completati, i file po generati verranno inseriti nelle seguenti directory:

- ./locale/de/LC_MESSAGES/
- ./locale/ja/LC_MESSAGES/

5. Tradurre i file po.

Come notato sopra, questi si trovano in ./locale/<lang>/LC_MESSAGES. Di seguito è riportato un esempio di uno di questi file, da Sphinx, builders.po.

```
# a5600c3d2e3d48fc8c261ea0284db79b
#: ../../builders.rst:4
msgid "Available builders"
msgstr "<FILL HERE BY TARGET LANGUAGE>"
```

Un altro caso, msgid è un testo su più righe e contiene la sintassi reStructuredText:

```
# 302558364e1d41c69b3277277e34b184
#: ../../builders.rst:9
msgid ""
"These are the built-in Sphinx builders. More builders can be
→added by "
":ref:`extensions <extensions>`."
msgstr ""
"FILL HERE BY TARGET LANGUAGE FILL HERE BY TARGET LANGUAGE FILL
→HERE "
"BY TARGET LANGUAGE :ref:`EXTENSIONS <extensions>` FILL HERE."
```

Fare attenzione a non interrompere la notazione reST. La maggior parte degli editori po saranno di aiuto.

6. Build del documento tradotto.

C'è bisogno di un parametro *language* in conf.py oppure si può anche specificare il parametro sulla riga di comando.

Per il make BSD/GNU, si esegue:

```
$ make -e SPHINXOPTS="-D language='de'" html
```

Per cmd.exe di Windows, si esegue:

```
> set SPHINXOPTS=-D language=de
> .\make.bat html
```

Per PowerShell, si esegue:

```
> Set-Item env:SPHINXOPTS "-D language=de"
> .\make.bat html
```

Congratulazioni! La documentazione tradotta è nella directory `_build/html`.

Added in version 1.3: **sphinx-build** invocato dal comando `make` creerà i file `mo` dai file `po`.

Se si usa la versione 1.2.x o precedente, si richiama il comando **sphinx-intl build** prima del comando **make**.

Traduzione

Aggiornare i file `po` con i nuovi file `pot`

Se un documento viene aggiornato, è necessario generare nuovi file `pot` e applicare le differenze ai file `po` tradotti. Per applicare gli aggiornamenti da un file `pot` al file `po`, utilizzare il comando **sphinx-intl update**.

```
$ sphinx-intl update -p _build/gettext
```

Utilizzo del servizio Transifex per team di traduzione

[Transifex³⁷³](#) è uno dei numerosi servizi che consentono la traduzione collaborativa tramite un'interfaccia web. Ha un elegante client a riga di comando basato su Go che semplifica il fetch e il push delle traduzioni.

1. Installare il [Transifex CLI tool³⁷⁴](#).

È necessario il tool da riga di comando **tx** per caricare le risorse (file `pot`). Il processo di installazione ufficiale inserisce il file binario **tx** nella directory corrente insieme a un file README e un file LICENSE e aggiunge la directory corrente a `$PATH`.

```
$ curl -o- https://raw.githubusercontent.com/transifex/cli/master/
  ↪ install.sh | bash
```

Vedi anche:

[Transifex Client documentation³⁷⁵](#)

2. Creare l'account [Transifex³⁷⁶](#) e creare un nuovo progetto e un'organizzazione per il documento.

Attualmente, Transifex non consente a un progetto di traduzione di avere più di una versione del documento, quindi è meglio includere un numero di versione nel nome del progetto.

Per esempio:

Organization ID
`sphinx-document`

³⁷³ <https://app.transifex.com/>

³⁷⁴ <https://github.com/transifex/cli/>

³⁷⁵ <https://developers.transifex.com/docs/using-the-client>

³⁷⁶ <https://app.transifex.com/>

Project ID

sphinx-document-test_1_0

Project URL

https://www.transifex.com/projects/p/sphinx-document-test_1_0/

3. Creare un token API da utilizzare nella riga di comando.

Nella pagina [Token API Transifex](#)³⁷⁷ generare un token. Copiare ora il token generato, poiché lo si potrà rivedere in seguito.

4. Impostare il token API Transifex nel file di configurazione utente \$HOME/.transifexrc.

```
[https://app.transifex.com]
rest_hostname = https://rest.api.transifex.com
token         = paste_your_api_token_here
```

5. In alternativa, si può memorizzare il token API Transifex come variabile di ambiente TX_TOKEN, che viene riconosciuta e utilizzata da **tx**.

```
$ export TX_TOKEN=paste_your_api_token_here
```

6. Creare il file di configurazione del progetto per il comando **tx**.

Questo processo creerà .tx/config nella directory corrente.

```
$ cd /your/document/root
$ tx init

Successful creation of '.tx/config' file
```

7. Caricamento dei file pot sul servizio Transifex.

Registrare i file pot nel file .tx/config utilizzando **sphinx-intl update-txconfig-resources**, adattando il valore --pot-dir alla directory dei file pot del progetto:

```
$ cd /your/document/root
$ sphinx-intl update-txconfig-resources --pot-dir _build/locale \
--transifex-organization-name=sphinx-document \
--transifex-project-name=sphinx-document-test_1_0
```

È possibile utilizzare le variabili di ambiente SPHINXINTL_TRANSIFEX_ORGANIZATION_NAME e SPHINXINTL_TRANSIFEX_PROJECT_NAME invece dei rispettivi argomenti della riga di comando.

Vedi anche:

[sphinx-intl update-txconfig-resources documentation](#)³⁷⁸

³⁷⁷ <https://app.transifex.com/user/settings/api/>

³⁷⁸ <https://sphinx-intl.readthedocs.io/en/master/refs.html#sphinx-intl-update-txconfig-resources>

e si caricano i file:

```
$ tx push -s
# Getting info about resources

sphinx-document-test_1_0.builders - Getting info
sphinx-document-test_1_0.builders - Done

# Pushing source files

sphinx-document-test_1_0.builders - Uploading file
sphinx-document-test_1_0.builders - Done
```

8. Inoltro della traduzione su Transifex.

9. Estrazione dei file po e creazione dell'HTML tradotto.

Dai cataloghi tradotti si creano i file mo. Per esempio, per creare i file mo per il tedesco (de):

```
$ cd /your/document/root
$ tx pull -l de
# Getting info about resources

sphinx-document-test_1_0.builders - Getting info
sphinx-document-test_1_0.builders - Done

# Pulling files

sphinx-document-test_1_0.builders [de] - Pulling file
sphinx-document-test_1_0.builders [de] - Creating download job
sphinx-document-test_1_0.builders [de] - Done
```

Invocare **make html** (per il BSD/GNU make) passando il codice della lingua:

```
$ make -e SPHINXOPTS="-D language='de'" html
```

È tutto!

Suggerimento: Tradurre localmente e su Transifex

Per eseguire il push dei file po di tutte le lingue, si può fare col comando **tx push -t**. Attenzione! Questa operazione sovrascrive le traduzioni in Transifex.

In altre parole, aggiornando sia il servizio che i file po locali, ci vorrebbe molto tempo e impegno per integrarli.

Utilizzo del servizio Transifex per team di traduzione

Maggiori informazioni nella [Weblate's documentation](#)³⁷⁹.

Contribuire alla traduzione dei reference di Sphinx

Il modo consigliato ai nuovi contributori di tradurre il reference è quello di unirsi al team di traduzione su Transifex.

C'è una [sphinx translation page](#)³⁸⁰ per la documentazione di Sphinx (master).

1. Login nel servizio Transifex³⁸¹.
2. Si va nella [sphinx translation page](#)³⁸².
3. Click su Request language e si compila il form.
4. Attendere l'accettazione da parte dei manutentori della traduzione di Transifex di sphinx.
5. (Dopo l'accettazione) Tradurre su Transifex.

Per i dettagli consultare: <https://help.transifex.com/en/articles/6248698-getting-started-as-a-translator>

Avanzamento e statistiche della traduzione

Added in version 7.1.0.

Durante il processo di rendering, Sphinx contrassegna ciascun nodo traducibile con un attributo `translated`, indicando se è stata trovata una traduzione per il testo in quel nodo.

Il valore di configurazione `translation_progress_classes` è utilizzabile per aggiungere una classe a ciascun elemento, a seconda del valore dell'attributo `translated`.

La sostituzione `|translation progress|` è utilizzabile per visualizzare la percentuale di nodi che sono stati tradotti per ciascun documento.

4.10 Supporto Web di Sphinx

Added in version 1.1.

Sphinx fornisce un'API Python per integrare facilmente la documentazione Sphinx nella propria applicazione web. Per saperne di più leggere il *Quick Start del Supporto Web*.

³⁷⁹ <https://docs.weblate.org/en/latest/develop/sphinx.html>

³⁸⁰ <https://app.transifex.com/sphinx-doc/sphinx-doc/>

³⁸¹ <https://app.transifex.com/>

³⁸² <https://app.transifex.com/sphinx-doc/sphinx-doc/>

Quick Start del Supporto Web

Building dei Dati della Documentazione

Per utilizzare il pacchetto di supporto Web nell'applicazione si dovranno creare i dati che utilizza. Questi dati includono file pickle che rappresentano documenti, indici di ricerca e dati dei nodi utilizzati per tenere traccia della posizione dei commenti e di altri elementi in un documento. Per farlo si dovrà creare un'istanza della classe `WebSupport` e chiamare il suo metodo `build()`:

```
from sphinxcontrib.websupport import WebSupport

support = WebSupport(sourcedir='/path/to/rst/sources/',
                     builddir='/path/to/build/outdir',
                     search='xapian')

support.build()
```

Questo leggerà i sorgenti reStructuredText da `sourcedir` e inserirà i dati necessari in `builddir`. La `builddir` conterrà due sottodirectory: una denominata "data" che contiene tutti i dati necessari per visualizzare documenti, effettuare ricerche tra i documenti e aggiungere commenti ai documenti. L'altra directory sarà chiamata "static" e conterrà file statici che dovrebbero essere serviti da "/static".

Nota: Per servire file statici da un percorso diverso da "/static", si deve fornire l'argomento parola chiave `staticdir` durante la creazione dell'oggetto `WebSupport`.

Integrazione dei Documenti Sphinx Nella Propria Webapp

Ora che i dati sono stati creati, è il momento di farne qualcosa di utile. Si inizia creando un oggetto `WebSupport` per l'applicazione:

```
from sphinxcontrib.websupport import WebSupport

support = WebSupport(datadir='/path/to/the/data',
                     search='xapian')
```

Ne servirà solo uno per ogni set di documentazione con cui si lavorerà. Poi si può chiamare il suo metodo `get_document()` per accedere ai singoli documenti:

```
contents = support.get_document('contents')
```

Questo restituirà un dizionario contenente i seguenti elementi:

- **body:** Il corpo principale del documento in formato HTML
- **sidebar:** La sidebar [*barra laterale*] del documento in formato HTML

- **relbar**: Un div contenente link a documenti correlati
- **title**: Il titolo del documento
- **css**: Link ai file CSS utilizzati da Sphinx
- **script**: JavaScript contenente le opzioni dei commenti

Questo dict può poi essere utilizzato come contesto per i template. L'obiettivo è quello di essere facilmente integrabile con il sistema di template esistente. Un esempio che utilizza Jinja2³⁸⁴ è:

```
{%- extends "layout.html" %}

{%- block title %}
    {{ document.title }}
{%- endblock %}

{% block css %}
    {{ super() }}
    {{ document.css|safe }}
    <link rel="stylesheet" href="/static/websupport-custom.css" type=
→ "text/css">
{% endblock %}

{%- block script %}
    {{ super() }}
    {{ document.script|safe }}
{%- endblock %}

{%- block relbar %}
    {{ document.relbar|safe }}
{%- endblock %}

{%- block body %}
    {{ document.body|safe }}
{%- endblock %}

{%- block sidebar %}
    {{ document.sidebar|safe }}
{%- endblock %}
```

³⁸⁴ <https://jinja.palletsprojects.com/>

Autenticazione

Per utilizzare determinate funzionalità come la votazione, deve essere possibile autenticare gli utenti. I dettagli dell'autenticazione vengono lasciati all'applicazione. Una volta che un utente è stato autenticato, si possono passare i dettagli dell'utente a determinati metodi *WebSupport* utilizzando gli argomenti delle parole chiave *username* e *moderator*. Il pacchetto di supporto web memorizzerà l'username con commenti e voti. L'unica avvertenza è che se si consente agli utenti di modificare il proprio username è necessario aggiornare i dati del pacchetto websupport:

```
support.update_username(old_username, new_username)
```

username dovrebbe essere una stringa univoca che identifica un utente e *moderator* dovrebbe essere un booleano che indica se l'utente ha privilegi di moderazione. Il valore di default per *moderator* è False.

Un esempio di funzione *Flask*³⁸⁵ che controlla se un utente ha effettuato l'accesso e poi recupera un documento è:

```
from sphinxcontrib.websupport.errors import *

@app.route('/<path:docname>')
def doc(docname):
    username = g.user.name if g.user else ''
    moderator = g.user.moderator if g.user else False
    try:
        document = support.get_document(docname, username, moderator)
    except DocumentNotFoundError:
        abort(404)
    return render_template('doc.html', document=document)
```

La prima cosa da notare è che *docname* è solo il percorso della richiesta. Questo semplifica l'accesso al documento corretto da un'unica vista. Se l'utente è autenticato, l'username e lo stato di moderazione vengono passati insieme al nome del documento a *get_document()*. Il pacchetto di supporto web aggiungerà quindi questi dati alle COMMENT_OPTIONS utilizzate nel template.

Nota: Funziona solo se la documentazione viene fornita dalla root dei documenti. Se viene servito da un'altra directory, si dovrà prefissare il percorso URL con quella directory e fornire parola chiave *docroot* come argomento durante la creazione dell'oggetto di supporto web:

```
support = WebSupport(..., docroot='docs')

@app.route('/docs/<path:docname>')
```

³⁸⁵ <https://flask.palletsprojects.com/>

Esecuzione delle Ricerche

Per utilizzare il modulo di ricerca nativo nella sidebar Sphinx, si crea una funzione per gestire le richieste all'URL 'search' relativo alla radice della documentazione. La query di ricerca dell'utente sarà nei parametri GET, con la chiave `q`. Poi si usa il metodo `get_search_results()` per recuperare i risultati della ricerca. In [Flask](https://flask.palletsprojects.com/)³⁸⁶ sarebbe così:

```
@app.route('/search')
def search():
    q = request.args.get('q')
    document = support.get_search_results(q)
    return render_template('doc.html', document=document)
```

Notare che per visualizzare i risultati della ricerca abbiamo utilizzato lo stesso template utilizzato per visualizzare i nostri documenti. Questo perché `get_search_results()` restituisce un comando di contesto nello stesso formato di `get_document()`.

Commenti & Proposte

Ora che questo è fatto è il momento di definire le funzioni che gestiscono le chiamate AJAX dallo script. Ci sarà bisogno di tre funzioni. La prima funzione viene utilizzata per aggiungere un nuovo commento e chiamerà il metodo del supporto web `add_comment()`:

```
@app.route('/docs/add_comment', methods=['POST'])
def add_comment():
    parent_id = request.form.get('parent', '')
    node_id = request.form.get('node', '')
    text = request.form.get('text', '')
    proposal = request.form.get('proposal', '')
    username = g.user.name if g.user is not None else 'Anonymous'
    comment = support.add_comment(text, node_id='node_id',
                                  parent_id='parent_id',
                                  username=username, proposal=proposal)
    return jsonify(comment=comment)
```

Si noterà che con la richiesta vengono inviati sia un `parent_id` che `node_id`. Se il commento viene allegato direttamente a un nodo, `parent_id` sarà vuoto. Se il commento è figlio di un altro commento, allora `node_id` sarà vuoto. La funzione successiva gestisce il recupero dei commenti per un nodo specifico ed è appropriatamente chiamata `get_data()`:

```
@app.route('/docs/get_comments')
def get_comments():
    username = g.user.name if g.user else None
    moderator = g.user.moderator if g.user else False
    node_id = request.args.get('node', '')
```

(continues on next page)

³⁸⁶ <https://flask.palletsprojects.com/>

(continua dalla pagina precedente)

```
data = support.get_data(node_id, username, moderator)
return jsonify(**data)
```

La funzione finale necessaria chiamerà `process_vote()` e gestirà i voti degli utenti sui commenti:

```
@app.route('/docs/process_vote', methods=['POST'])
def process_vote():
    if g.user is None:
        abort(401)
    comment_id = request.form.get('comment_id')
    value = request.form.get('value')
    if value is None or comment_id is None:
        abort(400)
    support.process_vote(comment_id, g.user.id, value)
    return "success"
```

Moderazione dei Commenti

Per default, tutti i commenti aggiunti tramite `add_comment()` vengono visualizzati automaticamente. Per avere una qualche forma di moderazione, si può passare l'argomento della parola chiave `displayed`:

```
comment = support.add_comment(text, node_id='node_id',
                              parent_id='parent_id',
                              username=username, proposal=proposal,
                              displayed=False)
```

È quindi possibile creare una nuova vista per gestire la moderazione dei commenti. Verrà chiamato quando un moderatore decide che un commento deve essere accettato e visualizzato:

```
@app.route('/docs/accept_comment', methods=['POST'])
def accept_comment():
    moderator = g.user.moderator if g.user else False
    comment_id = request.form.get('id')
    support.accept_comment(comment_id, moderator=moderator)
    return 'OK'
```

Il rifiuto dei commenti avviene tramite la loro cancellazione.

Per eseguire un'azione personalizzata (come inviare un'e-mail a un moderatore) quando un nuovo commento viene aggiunto ma non visualizzato, puoi passare callable alla classe `WebSupport` quando si istanzia il proprio oggetto di supporto:

```
def moderation_callback(comment):
    """Do something..."""
```

(continues on next page)

(continua dalla pagina precedente)

```
support = WebSupport(..., moderation_callback=moderation_callback)
```

La callback di moderazione deve accettare un argomento, che sarà lo stesso dict di commento restituito da `WebSupport.add_comment()`.

La Classe WebSupport

`class sphinxcontrib.websupport.WebSupport`

La classe API principale per il pacchetto di supporto Web. Tutte le interazioni con il pacchetto di supporto Web dovrebbero avvenire tramite questa classe.

La classe accetta i seguenti argomenti di parole chiave:

srcdir

La directory contenente i file sorgenti reStructuredText.

builddir

È necessario inserire la directory in cui creare dati e file statici. Dovrebbe essere usato quando si crea un oggetto `WebSupport` che verrà utilizzato per la build dei dati.

datadir

La directory in cui si trovano i dati del supporto Web. Dovrebbe essere utilizzato durante la creazione di un oggetto `WebSupport` che verrà utilizzato per recuperare i dati.

search

Questo può contenere una stringa (ad esempio 'xapian') che fa riferimento a un search adapter [*adattatore di ricerca*] nativo da utilizzare o un'istanza di una sottoclasse di `BaseSearch`.

storage

Può contenere una stringa che rappresenta l'URI di un database o un'istanza di una sottoclasse di `StorageBackend`. Se questo non viene fornito, verrà creato un nuovo database sqlite.

moderation_callback

Un callable da chiamare quando viene aggiunto un nuovo commento che non viene visualizzato. Deve accettare un argomento: un dizionario che rappresenta il commento aggiunto.

staticdir

Se i file statici devono essere creati in una posizione diversa **e non in** `'/static'`, questa dovrebbe essere una stringa con il nome di quella posizione (ad esempio `builddir + '/static_files'`).

Nota: Se si specifica `staticdir`, in genere si vorrà modificare `staticroot` di conseguenza.

staticroot

Se i file statici non vengono serviti da `'/static'`, questa dovrebbe essere una stringa con il nome di quella posizione (ad esempio `'/static_files'`).

docroot

Se la documentazione non viene fornita dal percorso di base di un URL, questa dovrebbe essere una stringa che specifica quel percorso (ad esempio `'docs'`).

Cambiato nella versione 1.6: La classe `WebSupport` viene spostata in `sphinxcontrib.websupport` da `sphinx.websupport`. Aggiungere il pacchetto `sphinxcontrib-websupport` nella dipendenza e usare invece la classe spostata.

Metodi

WebSupport.build()

Build della documentazione. Inserisce i dati nella directory *outdir*. Lo si usa in questo modo:

```
support = WebSupport(sourcedir, builddir, search='xapian')
support.build()
```

Questo leggerà i file di testo reStructured da *sourcedir*. Poi creerà i "pickle" e l'indice di ricerca, inserendoli in *builddir*. Inoltre salverà i dati del nodo nel database.

WebSupport.get_document(docname, username="", moderator=False)

Carica e restituisce un documento da un pickle. Il documento sarà un oggetto dict utilizzabile per eseguire il rendering di un template:

```
support = WebSupport(datadir=datadir)
support.get_document('index', username, moderator)
```

Nella maggior parte dei casi *docname* verrà preso dal percorso della richiesta e passato direttamente a questa funzione. In Flask, sarebbe qualcosa del genere:

```
@app.route('/<path:docname>')
def index(docname):
    username = g.user.name if g.user else ''
    moderator = g.user.moderator if g.user else False
    try:
        document = support.get_document(docname, username,
                                         moderator)
    except DocumentNotFoundError:
        abort(404)
    render_template('doc.html', document=document)
```

Il documento restituito contiene i seguenti elementi da utilizzare durante il rendering del template.

- **body**: Il corpo principale del documento in formato HTML
- **sidebar**: La sidebar [*barra laterale*] del documento in formato HTML
- **relbar**: Un div contenente link a documenti correlati
- **title**: Il titolo del documento
- **css**: Link ai file css utilizzati da Sphinx
- **script**: Javascript contenente opzioni di commento

Ciò solleva il problema `DocumentNotFoundError` se un documento corrispondente a *docname* non viene trovato.

Parametri

docname -- il nome del documento da caricare.

`WebSupport.get_data(node_id, username=None, moderator=False)`

Ottiene i commenti e il sorgente associati a *node_id*. Se viene fornito *username* le informazioni sul voto verranno incluse nei commenti restituiti. Il `CommentBackend` di default restituisce un dict con due chiavi, *source* e *comments*. *source* è l'origine grezza del nodo e viene utilizzata come punto di partenza per i suggerimenti che un utente può aggiungere. *comments* è una lista di dict che rappresentano un commento, ciascuno con i seguenti elementi:

Key [<i>chiave</i>]	Contenuti
text	Il testo di commento.
username	L'username memorizzato col commento.
id	L'identificatore univoco del commento.
rating	La valutazione attuale del commento.
age	Il tempo in secondi trascorso dall'aggiunta del commento.
time	Un dict contenente informazioni sull'ora. Contiene le seguenti chiavi: <i>year</i> , <i>month</i> , <i>day</i> , <i>hour</i> , <i>minute</i> , <i>second</i> , <i>iso</i> e <i>delta</i> . <i>iso</i> è l'ora formattata nel formato ISO 8601. <i>delta</i> è una forma stampabile dell'età del commento (ad esempio "3 ore fa").
vote	Se è stato fornito <i>user_id</i> , questo sarà un numero intero che rappresenta il voto. 1 per un voto positivo, -1 per un voto negativo o 0 se non votato.
node	L'ID del nodo a cui è allegato il commento. Se il genitore del commento è un altro commento anziché un nodo, questo sarà nullo.
parent	L'ID del commento a cui è allegato questo commento se non è allegato a un nodo.
children	Una lista di tutti i figli, in questo formato.
proposal_diff	Una rappresentazione HTML delle differenze tra il sorgente corrente e quella proposta dall'utente.

Parametri

- **node_id** -- l'id del nodo per cui ottenere i commenti.
- **username** -- l'username dell'utente che visualizza i commenti.
- **moderator** -- se l'utente è un moderatore.

```
WebSupport.add_comment(text, node_id="", parent_id="", displayed=True,
                       username=None, time=None, proposal=None,
                       moderator=False)
```

Aggiunge un commento a un nodo o a un altro commento. Restituisce il commento nello stesso formato di `get_comments()`. Se il commento viene allegato a un nodo, passa l'ID del nodo (come una stringa) con l'argomento della parola chiave `node`:

```
comment = support.add_comment(text, node_id=node_id)
```

Se il commento è figlio di un altro commento, si fornisce l'id del genitore (come stringa) con l'argomento della parola chiave del genitore:

```
comment = support.add_comment(text, parent_id=parent_id)
```

Per memorizzare un username con il commento, si passa l'argomento facoltativo della parola chiave `username`:

```
comment = support.add_comment(text, node=node_id,
                              username=username)
```

Parametri

- **parent_id** -- l'ID prefissato del genitore del commento.
- **text** -- il testo del commento.
- **displayed** -- a scopo di moderazione
- **username** -- l'username dell'utente che ha effettuato il commento.
- **time** -- l'ora in cui è stato creato il commento, per default è "now".

```
WebSupport.process_vote(comment_id, username, value)
```

Elabora il voto di un utente. Il pacchetto di supporto Web si basa sull'API utente per eseguire l'autenticazione. L'API utente in genere riceverà un `comment_id` e un valore da una form e poi si assicurerà che l'utente sia autenticato. È necessario inserire un username univoco, che verrà utilizzato anche per recuperare i dati delle votazioni passate dell'utente. Un esempio, ancora una volta in Flask:

```
@app.route('/docs/process_vote', methods=['POST'])
def process_vote():
    if g.user is None:
        abort(401)
    comment_id = request.form.get('comment_id')
    value = request.form.get('value')
```

(continues on next page)

(continua dalla pagina precedente)

```
if value is None or comment_id is None:
    abort(400)
support.process_vote(comment_id, g.user.name, value)
return "success"
```

Parametri

- **comment_id** -- il commento in fase di votazione
- **username** -- L'username univoco dell'utente votante
- **value** -- 1 per un voto positivo, -1 per un voto negativo, 0 per un voto negativo.

WebSupport.get_search_results(*q*)

Esegue una ricerca per la query *q* e crea una serie di risultati della ricerca. Poi visualizza i risultati della ricerca come html e restituisce un comando di contesto come quello creato da [get_document\(\)](#):

```
document = support.get_search_results(q)
```

Parametri

q -- la query di ricerca

I Search Adapter

Per creare un search adapter [*adattatore di ricerca*] personalizzato si dovrà creare una sottoclasse della classe [BaseSearch](#). Poi si crea un'istanza della nuova classe e la si passa come parola chiave *search* all'argomento quando si crea l'oggetto [WebSupport](#):

```
support = WebSupport(srcdir=srcdir,
                     builddir=builddir,
                     search=MySearch())
```

Per ulteriori informazioni sulla creazione di un *adattatore di ricerca* personalizzato, consultare la documentazione della classe [BaseSearch](#).

class sphinxcontrib.websupport.search.BaseSearch

Definisce un'interfaccia per i "search adapter".

Cambiato nella versione 1.6: La classe `BaseSearch` viene spostata in `sphinxcontrib.websupport.search` da `sphinx.websupport.search`.

Metodi

I seguenti metodi sono definiti nella classe `BaseSearch`. Alcuni metodi non hanno bisogno di essere sovrascritti, ma alcuni (`add_document()` e `handle_query()`) devono essere sovrascritti (overridden) nella propria sottoclasse. Per un esempio funzionante, guardare l'adapter nativo per whoosh.

`BaseSearch.init_indexing(changed=[])`

Chiamato dal builder per inizializzare l'indicizzatore della ricerca. *changed* è una lista di pagename che verranno reindicizzate. Si potrebbe volerli rimuovere dall'indice di ricerca prima che inizi l'indicizzazione.

Parametri

changed -- una lista di pagename che verranno reindicizzate

`BaseSearch.finish_indexing()`

Chiamato dal builder a scrittura ultimata. Usarlo per eseguire eventuali azioni di finalizzazione o pulizia una volta completata l'indicizzazione.

`BaseSearch.feed(pagename, filename, title, doctree)`

Chiamato dal builder per aggiungere un doctree all'indice. Converte il *doctree* in testo e lo passa a `add_document()`. Probabilmente non lo si sovrascriverà a meno che non ci sia bisogno di accedere al *doctree*. Override [sovrascrive] invece `add_document()`.

Parametri

- **pagename** -- il nome della pagina da indicizzare
- **filename** -- il nome del file sorgente originale
- **title** -- il titolo della pagina da indicizzare
- **doctree** -- è la rappresentazione del doctree della pagina

`BaseSearch.add_document(pagename, filename, title, text)`

Chiamato da `feed()` per aggiungere un documento all'indice di ricerca. Questo metodo dovrebbe fare tutto il necessario per aggiungere un singolo documento all'indice di ricerca.

pagename è il nome della pagina da indicizzare. È la combinazione del percorso relativo e del filename del file sorgente, meno l'estensione. Per esempio, se il file sorgente è "ext/builders.rst", il *pagename* sarà "ext/builders". Questo dovrà essere restituito con i risultati della ricerca durante l'elaborazione di una query.

Parametri

- **pagename** -- il nome della pagina da indicizzare
- **filename** -- il nome del file sorgente originale
- **title** -- il titolo della pagina
- **text** -- il testo completo della pagina

`BaseSearch.query(q)`

Chiamato dall'API di supporto Web per ottenere i risultati della ricerca. Questo metodo compila l'espressione regolare da utilizzare quando *si estrae il contesto*, poi

chiama `handle_query()`. Non lo si vorrà sovrascrivere (override) a meno che non si voglia utilizzare il metodo `extract_context()` incluso. Override [sovrascrive] invece `handle_query()`.

Parametri

q -- la stringa della query di ricerca.

`BaseSearch.handle_query(q)`

Chiamato da `query()` per recuperare i risultati della ricerca per una query di ricerca *q*. Questo dovrebbe restituire un iterabile contenente tuple del seguente formato:

```
(<path>, <title>, <context>)
```

path e *title* sono gli stessi valori passati a `add_document()`, e *context* dovrebbe essere un breve frammento di testo attorno alla query di ricerca nel documento.

Il metodo `extract_context()` viene fornito come un modo semplice per creare il *context*.

Parametri

q -- la query di ricerca

`BaseSearch.extract_context(text, length=240)`

Estrae il contesto per la query di ricerca dal *text* completo del documento.

Parametri

- **text** -- il testo completo del documento per cui creare il contesto
- **length** -- la lunghezza dello snippet del contesto da restituire.

Gli Storage Backend

Per creare un backend di archiviazione personalizzato si dovrà creare una sottoclasse della classe `StorageBackend`. Poi si crea un'istanza della nuova classe e la si passa come parola chiave *storage* all'argomento quando si crea l'oggetto `WebSupport`:

```
support = WebSupport(srcdir=srcdir,  
                    builddir=builddir,  
                    storage=MyStorage())
```

Per ulteriori informazioni sulla creazione di un backend di archiviazione personalizzato, consultare la documentazione della classe `StorageBackend` di seguito.

class sphinxcontrib.websupport.storage.StorageBackend

Definisce un'interfaccia per i backend di archiviazione.

Cambiato nella versione 1.6: La classe `StorageBackend` viene spostata in `sphinxcontrib.websupport.storage` da `sphinx.websupport.storage`.

Metodi

`StorageBackend.pre_build()`

Chiamato immediatamente prima dell'inizio del processo di build. Usarlo per preparare `StorageBackend` per l'aggiunta di nodi.

`StorageBackend.add_node(id, document, source)`

Aggiunge un nodo a `StorageBackend`.

Parametri

- **id** -- un ID univoco per il commento.
- **document** -- il nome del documento a cui appartiene il nodo.
- **source** -- il nome del file sorgente.

`StorageBackend.post_build()`

Chiamato dopo il completamento di una build. Lo si usa per finalizzare l'aggiunta di nodi, se necessario.

`StorageBackend.add_comment(text, displayed, username, time, proposal, node_id, parent_id, moderator)`

Chiamato quando viene aggiunto un commento.

Parametri

- **text** -- il testo del commento
- **displayed** -- se il commento deve essere visualizzato
- **username** -- il nome dell'utente che aggiunge il commento
- **time** -- un oggetto "date" con l'ora in cui è stato aggiunto il commento
- **proposal** -- il testo della proposta avanzata dall'utente
- **node_id** -- l'ID del nodo a cui viene aggiunto il commento
- **parent_id** -- l'ID del commento genitore di questo commento.
- **moderator** -- se l'utente che aggiunge il commento è un moderatore

`StorageBackend.delete_comment(comment_id, username, moderator)`

Elimina un commento.

Solleva `UserNotAuthorizedError` se `moderator` è `False` e `username` non corrisponde all'`username` nel commento.

Parametri

- **comment_id** -- L'ID del commento da eliminare.
- **username** -- L'username dell'utente che richiede l'eliminazione.
- **moderator** -- Se l'utente è un moderatore.

`StorageBackend.get_data(node_id, username, moderator)`

Chiamato per recuperare tutti i dati per un nodo. Questo dovrebbe restituire un dict con due chiavi, *source* e *comments* come descritto dal metodo `get_data()` method di [WebSupport](#).

Parametri

- **node_id** -- L'ID del nodo per cui ottenere i dati.
- **username** -- Il nome dell'utente che richiede i dati.
- **moderator** -- Se il richiedente è un moderatore.

`StorageBackend.process_vote(comment_id, username, value)`

Elabora un voto in corso. *value* sarà -1, 0, o 1.

Parametri

- **comment_id** -- L'ID del commento su cui si sta votando.
- **username** -- L'username dell'utente che ha espresso il voto.
- **value** -- Il valore del voto espresso.

`StorageBackend.update_username(old_username, new_username)`

Se a un utente è consentito modificare il proprio username, questo metodo dovrebbe essere chiamato in modo che non vi siano dati "stagnanti" nel sistema di archiviazione.

Parametri

- **old_username** -- L'username in corso di modifica.
- **new_username** -- In cosa verrà modificato l'username.

`StorageBackend.accept_comment(comment_id)`

Chiamato quando un moderatore accetta un commento. Dopo che il metodo è stato chiamato, il commento dovrebbe essere visualizzato a tutti gli utenti.

Parametri

- **comment_id** -- L'ID del commento accettato.

Estendere Sphinx

Questa guida ha lo scopo di fornire una rapida introduzione per coloro che desiderano sviluppare le proprie estensioni per Sphinx. Sphinx possiede notevoli capacità di estensibilità, inclusa la capacità di agganciarsi a quasi ogni punto del processo di build. Per usare semplicemente le estensioni Sphinx esistenti, fare riferimento a *Uso di Sphinx*. Per una discussione più dettagliata sull'interfaccia dell'estensione vedere *Le API di Sphinx*.

5.1 Tutorial

Estendere la sintassi con ruoli e direttive

Panoramica

La sintassi sia di reStructuredText che di MyST può essere estesa creando nuove **direttive** - per gli elementi a livello di blocco - e **ruoli** - per gli elementi in linea.

In questo tutorial estenderemo Sphinx per aggiungere:

- Un ruolo `hello`, che genererà semplicemente il testo `Hello {text}!`.
- Una direttiva `hello`, che produrrà semplicemente il testo `Hello {text}!`, come paragrafo.

Per questa estensione c'è bisogno di una conoscenza di base di Python e introdurremo anche aspetti delle API `docutils`³⁸⁷.

³⁸⁷ <https://docutils.sourceforge.io/>

Impostazione del progetto

Si può utilizzare un progetto Sphinx esistente o crearne uno nuovo utilizzando **sphinx-quickstart**.

Con questo aggiungeremo l'estensione al progetto, all'interno della cartella `source`:

1. Creare una cartella `_ext` in `source`
2. Creare un nuovo file Python nella cartella `_ext` chiamato `helloworld.py`

Ecco un esempio della struttura delle cartelle che si potrebbe ottenere:

```
├── source
│   ├── _ext
│   │   └── helloworld.py
│   ├── conf.py
│   └── index.rst
```

Scrivere l'estensione

Aprire `helloworld.py` e incollare al suo interno il seguente codice:

```
1 from __future__ import annotations
2
3 from docutils import nodes
4
5 from sphinx.application import Sphinx
6 from sphinx.util.docutils import SphinxDirective, SphinxRole
7 from sphinx.util.typing import ExtensionMetadata
8
9
10 class HelloRole(SphinxRole):
11     """A role to say hello!"""
12
13     def run(self) -> tuple[list[nodes.Node], list[nodes.system_
14 ↪message]]:
15         node = nodes.inline(text=f'Hello {self.text}!')
16         return [node], []
17
18 class HelloDirective(SphinxDirective):
19     """A directive to say hello!"""
20
21     required_arguments = 1
22
23     def run(self) -> list[nodes.Node]:
24         paragraph_node = nodes.paragraph(text=f'hello {self.
25 ↪arguments[0]}!')
```

(continues on next page)

(continua dalla pagina precedente)

```

25     return [paragraph_node]
26
27
28 def setup(app: Sphinx) -> ExtensionMetadata:
29     app.add_role('hello', HelloRole())
30     app.add_directive('hello', HelloDirective)
31
32     return {
33         'version': '0.1',
34         'parallel_read_safe': True,
35         'parallel_write_safe': True,
36     }

```

In questo esempio accadono alcune cose essenziali:

La classe role

Il nostro nuovo ruolo è dichiarato nella classe `HelloRole`.

```

1 class HelloRole(SphinxRole):
2     """A role to say hello!"""
3
4     def run(self) -> tuple[list[nodes.Node], list[nodes.system_
5     ↳message]]:
6         node = nodes.inline(text=f'Hello {self.text}!')
7         return [node], []

```

Questa classe estende la classe `SphinxRole`. La classe contiene un metodo `run`, che è un requisito per ogni ruolo. Contiene la logica principale del ruolo e restituisce una tupla contenente:

- un elenco di nodi docutils in-linea che devono essere elaborati da Sphinx.
- un elenco (facoltativo) di nodi di messaggi di sistema

La classe directive

La nostra nuova direttiva è dichiarata nella classe `HelloDirective`.

```

1 class HelloDirective(SphinxDirective):
2     """A directive to say hello!"""
3
4     required_arguments = 1
5
6     def run(self) -> list[nodes.Node]:
7         paragraph_node = nodes.paragraph(text=f'hello {self.

```

(continues on next page)

(continua dalla pagina precedente)

```

→arguments[0]}!')
    return [paragraph_node]

```

Questa classe estende la classe *SphinxDirective*. La classe contiene un metodo `run`, che è un requisito per ogni direttiva. Contiene la logica principale della direttiva e restituisce un elenco di nodi docutils a livello di blocco che devono essere elaborati da Sphinx. Contiene anche un attributo `required_arguments`, che indica a Sphinx quanti argomenti sono richiesti per la direttiva.

Cosa sono i nodi docutils?

Quando Sphinx analizza un documento, crea un "Abstract Syntax Tree" (AST) ("albero sintattico astratto"), di nodi che rappresentano il contenuto del documento in modo strutturato, generalmente indipendente da qualsiasi formato di input (rST, MyST, ecc.) o di output (HTML, LaTeX, ecc.). È un albero perché ogni nodo può avere nodi figli e così via:

```

<document>
  <paragraph>
    <text>
      Hello world!

```

Il pacchetto `docutils`³⁸⁸ fornisce molti `nodii built-in`³⁸⁹, per rappresentare diversi tipi di contenuto come testo, paragrafi, riferimenti, tabelle, ecc.

Ogni tipo di nodo generalmente accetta solo un insieme specifico di nodi figli diretti, ad esempio il nodo `document` dovrebbe contenere solo nodi "block-level", come `paragraph`, `section`, `table`, ecc., mentre il nodo `paragraph` dovrebbe contenere solo nodi "inline-level", come `text`, `emphasis`, `strong`, ecc.

Vedi anche:

La documentazione docutils su `creating directives`³⁹⁰ e su `creating roles`³⁹¹.

La funzione `setup`

Questa funzione è un requisito. La usiamo per inserire la nostra nuova direttiva in Sphinx.

```

def setup(app: Sphinx) -> ExtensionMetadata:
    app.add_role('hello', HelloRole())
    app.add_directive('hello', HelloDirective)

    return {

```

(continues on next page)

³⁸⁸ <https://docutils.sourceforge.io/>

³⁸⁹ <https://docutils.sourceforge.io/docs/ref/doctree.html>

³⁹⁰ <https://docutils.sourceforge.io/docs/howto/rst-directives.html>

³⁹¹ <https://docutils.sourceforge.io/docs/howto/rst-roles.html>

(continua dalla pagina precedente)

```
'version': '0.1',
'parallel_read_safe': True,
'parallel_write_safe': True,
}
```

La cosa più semplice che si può fare è chiamare i metodi `Sphinx.add_role()` e `Sphinx.add_directive()`, che è ciò che abbiamo fatto qui. Per questa particolare chiamata, il primo argomento è il nome del ruolo/direttiva stesso utilizzato in un file reST. In questo caso useremo `hello`. Per esempio:

```
Some intro text here...

.. hello:: world

Some text with a :hello:`world` role.
```

Restituiamo anche i *metadati dell'estensione* che indicano la versione della nostra estensione, insieme al fatto che è sicuro utilizzare l'estensione sia per la lettura che per la scrittura parallele.

Uso dell'estensione

L'estensione deve essere dichiarata nel file `conf.py` per far sì che Sphinx se ne accorga. Qui sono necessari due passaggi:

1. Aggiungere la directory `_ext` al `path Python`³⁹² utilizzando `sys.path.append`. Questo dovrebbe essere posizionato nella parte superiore del file.
2. Aggiornare o crea la lista `extensions` e aggiungere il nome del file di estensione all'elenco

Per esempio:

```
import os
import sys

sys.path.append(os.path.abspath("./_ext"))

extensions = ['helloworld']
```

Suggerimento: Poiché non abbiamo installato la nostra estensione come `pacchetto Python`³⁹³, dobbiamo modificare il `path di Python`³⁹⁴ in modo che Sphinx possa trovare l'estensione. Questo è il motivo per cui abbiamo bisogno della chiamata a `sys.path.append`.

Ora si può utilizzare l'estensione in un file. Per esempio:

³⁹² <https://docs.python.org/3/using/cmdline.html#envvar-PYTHONPATH>

³⁹³ <https://packaging.python.org/>

³⁹⁴ <https://docs.python.org/3/using/cmdline.html#envvar-PYTHONPATH>

```
Some intro text here...
```

```
.. hello:: world
```

```
Some text with a :hello:`world` role.
```

L'esempio sopra genererebbe:

```
Some intro text here...
```

```
Hello world!
```

```
Some text with a hello world! role.
```

Ulteriori letture

Questo è il principio di un'estensione molto semplice che crea un nuovo ruolo e una nuova direttiva.

Per un esempio più avanzato, fare riferimento a *Estendere il processo di build*.

Per condividere la propria estensione su più progetti o con altri, consultare la sezione *Estensioni di terze parti*.

Estendere il processo di build

L'obiettivo di questo tutorial è creare un'estensione più completa di quella creata in *Estendere la sintassi con ruoli e direttive*. Mentre quella guida riguardava solo la scrittura di *role* e *directive* personalizzati, questa guida copre un'estensione più complessa al processo di build di Sphinx; aggiungendo più direttive, oltre a dei nodi personalizzati, ulteriori valori di configurazione e gestori di eventi personalizzati.

A tal fine, tratteremo un'estensione `todo` che aggiunge funzionalità per includere voci di cose da fare nella documentazione e per raccoglierle in un luogo centrale. Questo è simile all'estensione *sphinx.ext.todo* distribuita con Sphinx.

Panoramica

Nota: Per comprendere il design di questa estensione, fare riferimento a *Oggetti importanti* e a *Fasi della Build*.

Vogliamo che l'estensione aggiunga quanto segue a Sphinx:

- Una direttiva `todo`, con del contenuto contrassegnato con "TODO" e mostrato nell'output solo se viene impostato un nuovo valore di configurazione. Le voci `todo` on dovrebbero stare nell'output per default.

- Una direttiva `todo` che crea un elenco di tutte le voci `todo` in tutta la documentazione.

Per questo, dovremo aggiungere i seguenti elementi a Sphinx:

- Nuove direttive, chiamate `todo` e `todo`list.
- Nuovi nodi dell'albero dei documenti per rappresentare queste direttive, convenzionalmente chiamati anche `todo` e `todo`list. Non avremmo bisogno di nuovi nodi se le nuove direttive producessero solo del contenuto rappresentabile dai nodi esistenti.
- Un nuovo valore di configurazione `todo_include_todos` (i nomi dei valori di configurazione dovrebbero iniziare con il nome dell'estensione, per rimanere univoci) che controlla se le voci `todo` vengono inserite nell'output.
- Nuovi event handler [*gestori di eventi*]: uno per l'evento *doctree-resolved*, per sostituire i nodi `todo` e `todo`list, uno per *env-merge-info* per unire i risultati intermedi dalle build parallele e uno per *env-purge-doc* (il motivo verrà spiegato più avanti).

Prerequisiti

Come con *Estendere la sintassi con ruoli e direttive*, non distribuiremo questo plugin tramite PyPI quindi ancora una volta abbiamo bisogno di un progetto Sphinx da cui richiamarlo. Si può utilizzare un progetto esistente o crearne uno nuovo utilizzando **sphinx-quickstart**.

Supponiamo che si stia utilizzando cartelle dei sorgenti (`source`) e di build (`build`) separate. Il file di estensione potrebbe trovarsi in qualsiasi cartella del progetto. Nel nostro caso, facciamo quanto segue:

1. Creare una cartella `_ext` in `source`
2. Creare un nuovo file Python nella cartella `_ext` chiamata `todo.py`

Ecco un esempio della struttura delle cartelle che si potrebbe ottenere:

```
├── source
│   ├── _ext
│   │   └── todo.py
│   ├── _static
│   ├── conf.py
│   ├── somefolder
│   ├── index.rst
│   ├── somefile.rst
│   └── someotherfile.rst
```

Scrivere l'estensione

Aprire `todo.py` e incollare al suo interno il seguente codice, che spiegheremo in dettaglio a breve:

```
1 from docutils import nodes
2 from docutils.parsers.rst import Directive
3
4 from sphinx.application import Sphinx
5 from sphinx.locale import _
6 from sphinx.util.docutils import SphinxDirective
7 from sphinx.util.typing import ExtensionMetadata
8
9
10 class todo(nodes.Admonition, nodes.Element):
11     pass
12
13
14 class todolist(nodes.General, nodes.Element):
15     pass
16
17
18 def visit_todo_node(self, node):
19     self.visit_admonition(node)
20
21
22 def depart_todo_node(self, node):
23     self.depart_admonition(node)
24
25
26 class TodolistDirective(Directive):
27     def run(self):
28         return [todolist('')]
29
30
31 class TodoDirective(SphinxDirective):
32     # this enables content in the directive
33     has_content = True
34
35     def run(self):
36         targetid = 'todo-%d' % self.env.new_serialno('todo')
37         targetnode = nodes.target('', '', ids=[targetid])
38
39         todo_node = todo('\n'.join(self.content))
40         todo_node += nodes.title(_('Todo'), _('Todo'))
41         todo_node += self.parse_content_to_nodes()
42
```

(continues on next page)

(continua dalla pagina precedente)

```

43     if not hasattr(self.env, 'todo_all_todos'):
44         self.env.todo_all_todos = []
45
46     self.env.todo_all_todos.append({
47         'docname': self.env.docname,
48         'lineno': self.lineno,
49         'todo': todo_node.deepcopy(),
50         'target': targetnode,
51     })
52
53     return [targetnode, todo_node]
54
55
56 def purge_todos(app, env, docname):
57     if not hasattr(env, 'todo_all_todos'):
58         return
59
60     env.todo_all_todos = [todo for todo in env.todo_all_todos if todo[
61 ↪ 'docname'] != docname]
62
63 def merge_todos(app, env, docnames, other):
64     if not hasattr(env, 'todo_all_todos'):
65         env.todo_all_todos = []
66     if hasattr(other, 'todo_all_todos'):
67         env.todo_all_todos.extend(other.todo_all_todos)
68
69
70 def process_todo_nodes(app, doctree, fromdocname):
71     if not app.config.todo_include_todos:
72         for node in doctree.findall(todo):
73             node.parent.remove(node)
74
75     # Replace all todolist nodes with a list of the collected todos.
76     # Augment each todo with a backlink to the original location.
77     env = app.builder.env
78
79     if not hasattr(env, 'todo_all_todos'):
80         env.todo_all_todos = []
81
82     for node in doctree.findall(todolist):
83         if not app.config.todo_include_todos:
84             node.replace_self([])
85             continue
86
87     content = []

```

(continues on next page)

(continua dalla pagina precedente)

```

88
89     for todo_info in env.todo_all_todos:
90         para = nodes.paragraph()
91         filename = env.doc2path(todo_info['docname'], base=None)
92         description = _(
93             '(The original entry is located in %s, line %d and can
→be found '
94             ) % (filename, todo_info['lineno'])
95         para += nodes.Text(description)
96
97         # Create a reference
98         newnode = nodes.reference('', '')
99         innernode = nodes.emphasis(_('here'), _('here'))
100         newnode['refdocname'] = todo_info['docname']
101         newnode['refuri'] = app.builder.get_relative_
→uri(fromdocname, todo_info['docname'])
102         newnode['refuri'] += '#' + todo_info['target']['refid']
103         newnode.append(innernode)
104         para += newnode
105         para += nodes.Text('.')
106
107         # Insert into the todoclist
108         content.extend((
109             todo_info['todo'],
110             para,
111         ))
112
113         node.replace_self(content)
114
115
116 def setup(app: Sphinx) -> ExtensionMetadata:
117     app.add_config_value('todo_include_todos', False, 'html')
118
119     app.add_node(todolist)
120     app.add_node(
121         todo,
122         html=(visit_todo_node, depart_todo_node),
123         latex=(visit_todo_node, depart_todo_node),
124         text=(visit_todo_node, depart_todo_node),
125     )
126
127     app.add_directive('todo', TodoDirective)
128     app.add_directive('todolist', TodolistDirective)
129     app.connect('doctree-resolved', process_todo_nodes)
130     app.connect('env-purge-doc', purge_todos)
131     app.connect('env-merge-info', merge_todos)

```

(continues on next page)

(continua dalla pagina precedente)

```
132
133     return {
134         'version': '0.1',
135         'env_version': 1,
136         'parallel_read_safe': True,
137         'parallel_write_safe': True,
138     }
```

Si tratta di un'estensione molto più vasta di quella descritta in *Estendere la sintassi con ruoli e direttive*, tuttavia, esamineremo ogni parte passo dopo passo per spiegare cosa sta succedendo.

Le classi dei nodi

Cominciamo con le classi dei nodi:

```
1
2
3 class todo(nodes.Admonition, nodes.Element):
4     pass
5
6
7 class todolist(nodes.General, nodes.Element):
8     pass
9
10
11 def visit_todo_node(self, node):
12     self.visit_admonition(node)
13
14
```

Le classi dei nodi solitamente non devono fare altro che ereditare dalle classi docutils standard definite in `docutils.nodes`. `todo` eredita da `Admonition` perché dovrebbe essere gestito come una nota o un warning, `todolist` è solo un nodo "general".

Nota: Molte estensioni non dovranno creare le proprie classi di nodi e funzioneranno correttamente con i nodi già forniti da `docutils`³⁹⁵ e da *Sphinx*.

Attenzione: È importante sapere che mentre si può estendere Sphinx senza lasciare il proprio `conf.py`, se si dichiara un nodo ereditato proprio lì, si incapperà in un `PickleError`³⁹⁶ non ovvio. Quindi, se qualcosa va storto, assicurarsi di inserire i nodi ereditati in un modulo Python separato.

Per maggiori dettagli vedere:

³⁹⁵ <https://docutils.sourceforge.io/docs/ref/doctree.html>

- <https://github.com/sphinx-doc/sphinx/issues/6751>
- <https://github.com/sphinx-doc/sphinx/issues/1493>
- <https://github.com/sphinx-doc/sphinx/issues/1424>

Le classi direttive

Una classe direttiva è una classe che deriva solitamente da `docutils.parsers.rst.Directive`. L'interfaccia direttiva è trattata in dettaglio anche nella [documentazione docutils](#)³⁹⁷; l'importante è che la classe abbia attributi che configurino il markup consentito e un metodo `run` che restituisca un elenco di nodi.

Osservando innanzitutto la direttiva `TodolistDirective`:

```
1
2
3 class TodolistDirective(Directive):
4     def run(self):
```

È molto semplice, creare e restituire un'istanza della classe del nodo `todo`. La stessa direttiva `TodolistDirective` non ha né contenuto né argomenti che debbano essere gestiti. Questo ci porta alla direttiva `TodoDirective`:

```
1
2 class TodoDirective(SphinxDirective):
3     # this enables content in the directive
4     has_content = True
5
6     def run(self):
7         targetid = 'todo-%d' % self.env.new_serialno('todo')
8         targetnode = nodes.target('', '', ids=[targetid])
9
10        todo_node = todo('\n'.join(self.content))
11        todo_node += nodes.title(_('Todo'), _('Todo'))
12        todo_node += self.parse_content_to_nodes()
13
14        if not hasattr(self.env, 'todo_all_todos'):
15            self.env.todo_all_todos = []
16
17        self.env.todo_all_todos.append({
18            'docname': self.env.docname,
19            'lineno': self.lineno,
20            'todo': todo_node.deepcopy(),
21            'target': targetnode,
```

(continues on next page)

³⁹⁶ <https://docs.python.org/3/library/pickle.html#pickle.PickleError>

³⁹⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html>

(continua dalla pagina precedente)

```

22     })
23
24     return [targetnode, todo_node]
```

Qui vengono trattate diverse cose importanti. Innanzitutto, come si può vedere, stiamo creando una sottoclasse della classe helper *SphinxDirective* invece della solita classe *Directive*. Questo ci dà accesso all'*istanza dell'ambiente di build* utilizzando la proprietà `self.env`. Senza questo, dovremmo usare l'alquanto contorto `self.state.document.settings.env`. Quindi, per agire come target del link (da *TodolistDirective*), la direttiva *TodoDirective* deve restituire un nodo di target oltre al nodo *todo*. L'ID del target (in HTML, questo sarà il nome dell'ancora) viene generato utilizzando `env.new_serialno` che restituisce un nuovo intero univoco su ogni chiamata e quindi porta a nomi di target univoci. Il nodo di target viene istanziato senza testo (i primi due argomenti).

Durante la creazione del nodo di ammonizione, il corpo del contenuto della direttiva viene analizzato utilizzando `self.state.nested_parse`. Il primo argomento fornisce il corpo del contenuto e il secondo ne fornisce l'offset. Il terzo argomento fornisce il nodo genitore del risultato analizzato, nel nostro caso il nodo *todo*. Successivamente, il nodo *todo* viene aggiunto all'environment. Ciò è necessario per poter creare una lista di tutte le voci *todo* in tutta la documentazione, nel punto in cui l'autore inserisce una direttiva *todolist*. In questo caso, viene utilizzato l'attributo d'ambiente `todo_all_todos` (di nuovo, il nome dovrebbe essere univoco, quindi è preceduto dal nome dell'estensione). Non esiste quando viene creato un nuovo ambiente, quindi la direttiva deve verificarlo e crearlo se necessario. Varie informazioni sulla posizione della voce del *todo* vengono archiviate insieme a una copia del nodo.

Nell'ultima riga vengono restituiti i nodi che dovrebbero essere inseriti nel doctree: il nodo *target* e il nodo *ammonition*.

La struttura del nodo restituita dalla direttiva è simile alla seguente:

```

+-----+
| target node |
+-----+
+-----+
| todo node   |
+-----+
\__+-----+
   | admonition title |
   +-----+
   | paragraph         |
   +-----+
   | ...               |
   +-----+
```

I gestori di eventi

I gestori di eventi sono una delle funzionalità più potenti di Sphinx, poiché forniscono un modo per agganciarsi [hook] a qualsiasi parte del processo di documentazione. Ci sono molti eventi forniti da Sphinx stesso, come dettagliato nella *Guida alle API*, e qui ne utilizzeremo un sottoinsieme.

Diamo un'occhiata ai gestori di eventi utilizzati nell'esempio precedente. Innanzitutto, quello per l'evento *env-purge-doc*:

```

1 def purge_todos(app, env, docname):
2     if not hasattr(env, 'todo_all_todos'):
3         return
4
5     env.todo_all_todos = [todo for todo in env.todo_all_todos if todo[
6     ↪ 'docname'] != docname]
```

Poiché memorizziamo le informazioni dai file sorgenti nell'ambiente, che è persistente, potrebbero diventare obsolete quando il file sorgente cambia. Pertanto, prima che ciascun file sorgente venga letto, i relativi record dell'ambiente vengono cancellati e l'evento *env-purge-doc* offre alle estensioni la possibilità di fare lo stesso. Qui eliminiamo tutte le cose da fare il cui docname corrisponde a quello indicato dalla lista `todo_all_todos`. Se nel documento sono rimaste delle cose da fare, queste verranno aggiunte nuovamente durante l'analisi.

Il gestore successivo, per l'evento *env-merge-info*, viene utilizzato durante le build parallele. Poiché durante le build parallele tutti i thread hanno il proprio `env`, ci sono più liste `todo_all_todos` che devono essere riunite:

```

1 if not hasattr(env, 'todo_all_todos'):
2     env.todo_all_todos = []
3 if hasattr(other, 'todo_all_todos'):
4     env.todo_all_todos.extend(other.todo_all_todos)
5
```

L'altro gestore appartiene all'evento *doctree-resolved*:

```

1 if not app.config.todo_include_todos:
2     for node in doctree.findall(todo):
3         node.parent.remove(node)
4
5     # Replace all todoclist nodes with a list of the collected todos.
6     # Augment each todo with a backlink to the original location.
7     env = app.builder.env
8
9     if not hasattr(env, 'todo_all_todos'):
10         env.todo_all_todos = []
11
12     for node in doctree.findall(todolist):
```

(continues on next page)

(continua dalla pagina precedente)

```

13     if not app.config.todo_include_todos:
14         node.replace_self([])
15         continue
16
17     content = []
18
19     for todo_info in env.todo_all_todos:
20         para = nodes.paragraph()
21         filename = env.doc2path(todo_info['docname'], base=None)
22         description = _(
23             '(The original entry is located in %s, line %d and can
→be found '
24             ) % (filename, todo_info['lineno'])
25         para += nodes.Text(description)
26
27         # Create a reference
28         newnode = nodes.reference('', '')
29         innernode = nodes.emphasis(_('here'), _('here'))
30         newnode['refdocname'] = todo_info['docname']
31         newnode['refuri'] = app.builder.get_relative_
→uri(fromdocname, todo_info['docname'])
32         newnode['refuri'] += '#' + todo_info['target']['refid']
33         newnode.append(innernode)
34         para += newnode
35         para += nodes.Text('.')
36
37         # Insert into the todolist
38         content.extend((
39             todo_info['todo'],
40             para,
41         ))
42
43     node.replace_self(content)

```

L'evento *doctree-resolved* viene emesso alla fine della *fase 3 (risoluzione)* e consente di eseguire una risoluzione personalizzata. Il gestore che abbiamo scritto per questo evento è un po' più coinvolto. Se il valore di configurazione `todo_include_todos` (che descriveremo a breve) è falso, tutti i nodi `todo` e `todolist` vengono rimossi dai documenti. In caso contrario, i nodi `todo` rimangono dove e come sono. I nodi `todolist` vengono sostituiti da un elenco di voci di cose da fare, complete di backlink alla posizione da cui provengono. Le voci della lista sono composte dai nodi della voce `todo` e dai nodi `docutils` creati al volo: un paragrafo per ogni voce, contenente il testo che riporta la posizione, e un link (nodo di riferimento contenente un nodo corsivo) con la backreference. L'URI di riferimento è creato da *sphinx.builders.Builder.get_relative_uri()* che crea un URI adatto a seconda del builder utilizzato e aggiungendo l'ID del nodo `todo` (del target) come nome dell'ancora.

La funzione setup

Come notato *in precedenza*, la funzione `setup` è un requisito e viene utilizzata per inserire direttive in Sphinx. Tuttavia, la usiamo anche per agganciare le altre parti della nostra estensione. Diamo un'occhiata alla nostra funzione `setup`:

```

1 def setup(app: Sphinx) -> ExtensionMetadata:
2     app.add_config_value('todo_include_todos', False, 'html')
3
4     app.add_node(todolist)
5     app.add_node(
6         todo,
7         html=(visit_todo_node, depart_todo_node),
8         latex=(visit_todo_node, depart_todo_node),
9         text=(visit_todo_node, depart_todo_node),
10    )
11
12    app.add_directive('todo', TodoDirective)
13    app.add_directive('todolist', TodolistDirective)
14    app.connect('doctree-resolved', process_todo_nodes)
15    app.connect('env-purge-doc', purge_todos)
16    app.connect('env-merge-info', merge_todos)
17
18    return {
19        'version': '0.1',
20        'env_version': 1,
21        'parallel_read_safe': True,
22        'parallel_write_safe': True,
23    }
```

Le chiamate in questa funzione si riferiscono alle classi e alle funzioni aggiunte in precedenza. Ciò che fanno le singole chiamate è quanto segue:

- `add_config_value()` fa sapere a Sphinx che dovrebbe riconoscere il nuovo *valore di configurazione* `todo_include_todos`, il cui valore di default dovrebbe essere `False` (questo dice anche a Sphinx che si tratta di un valore booleano).

Se il terzo argomento fosse `'html'`, i documenti HTML verrebbero ricostruiti completamente se il "valore di configurazione" cambiasse il suo valore. Ciò è necessario per i valori di configurazione che influenzano la lettura (*build fase 1 (lettura)*).

- `add_node()` aggiunge una nuova *node class* al sistema di build. Si possono anche specificare funzioni visitor per ciascun formato di output supportato. Queste funzioni visitor sono necessarie quando i nuovi nodi rimangono fino alla *fase 4 (scrittura)*. Dato che il nodo `todolist` viene sempre sostituito nella *fase 3 (risoluzione)*, non ne ha bisogno.
- `add_directive()` aggiunge una nuova *direttiva*, data dal nome e dalla classe.
- Infine, `connect()` aggiunge un *event handler* all'evento il cui nome è dato dal primo argomento. La funzione del gestore eventi viene chiamata con diversi argomenti documentati con l'evento.

Con questo la nostra estensione è completa.

Uso dell'estensione

Come prima, dobbiamo abilitare l'estensione dichiarandola nel nostro file `conf.py`. Qui sono necessari due passaggi:

1. Aggiungere la directory `_ext` al `path Python`³⁹⁸ utilizzando `sys.path.append`. Questo dovrebbe essere posizionato nella parte superiore del file.
2. Aggiornare o crea la lista `extensions` e aggiungere il nome del file di estensione all'elenco

Inoltre, potremmo voler impostare il valore di configurazione `todo_include_todos`. Come notato sopra, il valore di default è `False` ma possiamo impostarlo esplicitamente.

Per esempio:

```
import os
import sys

sys.path.append(os.path.abspath("./_ext"))

extensions = ['todo']

todo_include_todos = False
```

Ora si può utilizzare l'estensione in tutto il progetto. Per esempio:

Listato 1: index.rst

```
Hello, world
=====

.. toctree::
   somefile.rst
   someotherfile.rst

Hello world. Below is the list of TODOs.

.. todolist::
```

Listato 2: somefile.rst

```
foo
===

Some intro text here...
```

(continues on next page)

³⁹⁸ <https://docs.python.org/3/using/cmdline.html#envvar-PYTHONPATH>

(continua dalla pagina precedente)

```
.. todo:: Fix this
```

Listato 3: someotherfile.rst

```
bar
```

```
===
```

```
Some more text here...
```

```
.. todo:: Fix that
```

Poiché abbiamo configurato `todo_include_todos` a `False`, in realtà non vedremo nulla visualizzato per le direttive `todo` e `todoist`. Tuttavia, se lo impostiamo su `true`, vedremo l'output descritto in precedenza.

Ulteriori letture

Per ulteriori informazioni, fare riferimento alla documentazione [docutils](#)³⁹⁹ e *Le API di Sphinx*.

Per condividere la propria estensione su più progetti o con altri, consultare la sezione *Estensioni di terze parti*.

Aggiungere un dominio di riferimento

L'obiettivo di questo tutorial è illustrare ruoli, direttive e domini. Una volta completato, saremo in grado di utilizzare questa estensione per descrivere una ricetta e fare riferimento a quella ricetta da qualche altra parte nella nostra documentazione.

Nota: Questo tutorial si basa su una guida pubblicata per la prima volta su [opensource.com](#)⁴⁰⁰ ed è fornita qui con il permesso dell'autore.

Panoramica

Vogliamo che l'estensione aggiunga quanto segue a Sphinx:

- Una direttiva *direttiva ricetta*, ha alcuni contenuti che descrivono i passaggi della ricetta, insieme a un'opzione `:contains:` *[contiene]* che evidenzia gli ingredienti principali della ricetta.
- Un *ruolo ref*, che fornisce un riferimento incrociato alla ricetta stessa.

³⁹⁹ <https://docutils.sourceforge.io/docs/>

⁴⁰⁰ <https://opensource.com/article/18/11/building-custom-workflows-sphinx>

- Un *dominio ricetta*, che ci consente di collegare insieme il ruolo e il dominio di cui sopra, insieme a cose come gli indici.

Per questo, dovremo aggiungere i seguenti elementi a Sphinx:

- Una nuova direttiva chiamata *ricetta*
- Nuovi indici per permetterci di fare riferimento a ingredienti e ricette
- Un nuovo dominio chiamato *ricetta*, che conterrà la direttiva *ricetta* e il ruolo *ref*

Prerequisiti

Abbiamo bisogno della stessa configurazione *delle estensioni precedenti*. Questa volta inseriremo l'estensione in un file chiamato `recipe.py`.

Ecco un esempio della struttura delle cartelle che si potrebbe ottenere:

```
├── source
│   ├── _ext
│   │   └── recipe.py
│   ├── conf.py
│   └── index.rst
```

Scrivere l'estensione

Aprire `recipe.py` e incollare al suo interno il seguente codice, che spiegheremo in dettaglio a breve:

```
1 from collections import defaultdict
2
3 from docutils.parsers.rst import directives
4
5 from sphinx import addnodes
6 from sphinx.application import Sphinx
7 from sphinx.directives import ObjectDescription
8 from sphinx.domains import Domain, Index
9 from sphinx.roles import XRefRole
10 from sphinx.util.nodes import make_refnode
11 from sphinx.util.typing import ExtensionMetadata
12
13
14 class RecipeDirective(ObjectDescription):
15     """A custom directive that describes a recipe."""
16
17     has_content = True
18     required_arguments = 1
19     option_spec = {
```

(continues on next page)

(continua dalla pagina precedente)

```

20     'contains': directives.unchanged_required,
21 }
22
23 def handle_signature(self, sig, signode):
24     signode += addnodes.desc_name(text=sig)
25     return sig
26
27 def add_target_and_index(self, name_cls, sig, signode):
28     signode['ids'].append('recipe' + '-' + sig)
29     if 'contains' in self.options:
30         ingredients = [x.strip() for x in self.options.get(
→ 'contains').split(',')]
31
32         recipes = self.env.get_domain('recipe')
33         recipes.add_recipe(sig, ingredients)
34
35
36 class IngredientIndex(Index):
37     """A custom index that creates an ingredient matrix."""
38
39     name = 'ingredient'
40     localname = 'Ingredient Index'
41     shortname = 'Ingredient'
42
43     def generate(self, docnames=None):
44         content = defaultdict(list)
45
46         recipes = {
47             name: (dispname, typ, docname, anchor)
48             for name, dispname, typ, docname, anchor, _ in self.domain.
→ get_objects()
49         }
50         recipe_ingredients = self.domain.data['recipe_ingredients']
51         ingredient_recipes = defaultdict(list)
52
53         # flip from recipe_ingredients to ingredient_recipes
54         for recipe_name, ingredients in recipe_ingredients.items():
55             for ingredient in ingredients:
56                 ingredient_recipes[ingredient].append(recipe_name)
57
58         # convert the mapping of ingredient to recipes to produce the
→ expected
59         # output, shown below, using the ingredient name as a key to
→ group
60         #
61         # name, subtype, docname, anchor, extra, qualifier, description

```

(continues on next page)

(continua dalla pagina precedente)

```

62     for ingredient, recipe_names in ingredient_recipes.items():
63         for recipe_name in recipe_names:
64             dispname, typ, docname, anchor = recipes[recipe_name]
65             content[ingredient].append((dispname, 0, docname,
→ anchor, docname, '', typ))
66
67         # convert the dict to the sorted list of tuples expected
68         content = sorted(content.items())
69
70     return content, True
71
72
73 class RecipeIndex(Index):
74     """A custom index that creates an recipe matrix."""
75
76     name = 'recipe'
77     localname = 'Recipe Index'
78     shortname = 'Recipe'
79
80     def generate(self, docnames=None):
81         content = defaultdict(list)
82
83         # sort the list of recipes in alphabetical order
84         recipes = self.domain.get_objects()
85         recipes = sorted(recipes, key=lambda recipe: recipe[0])
86
87         # generate the expected output, shown below, from the above,
→ using the
88         # first letter of the recipe as a key to group thing
89         #
90         # name, subtype, docname, anchor, extra, qualifier, description
91         for _name, dispname, typ, docname, anchor, _priority in
→ recipes:
92             content[dispname[0].lower()].append((
93                 dispname,
94                 0,
95                 docname,
96                 anchor,
97                 docname,
98                 '',
99                 typ,
100             ))
101
102         # convert the dict to the sorted list of tuples expected
103         content = sorted(content.items())
104

```

(continues on next page)

(continua dalla pagina precedente)

```

105     return content, True
106
107
108 class RecipeDomain(Domain):
109     name = 'recipe'
110     label = 'Recipe Sample'
111     roles = {
112         'ref': XRefRole(),
113     }
114     directives = {
115         'recipe': RecipeDirective,
116     }
117     indices = {
118         RecipeIndex,
119         IngredientIndex,
120     }
121     initial_data = {
122         'recipes': [], # object list
123         'recipe_ingredients': {}, # name -> object
124     }
125     data_version = 0
126
127     def get_full_qualified_name(self, node):
128         return f'recipe.{node.arguments[0]}'
129
130     def get_objects(self):
131         yield from self.data['recipes']
132
133     def resolve_xref(self, env, fromdocname, builder, typ, target,
134 ↪node, contnode):
135         match = [
136             (docname, anchor)
137             for name, sig, typ, docname, anchor, prio in self.get_
138 ↪objects()
139             if sig == target
140         ]
141
142         if len(match) > 0:
143             todocname = match[0][0]
144             targ = match[0][1]
145
146             return make_refnode(builder, fromdocname, todocname, targ,
147 ↪contnode, targ)
148         else:
149             print('Awww, found nothing')
150             return None

```

(continues on next page)

(continua dalla pagina precedente)

```

148
149     def add_recipe(self, signature, ingredients):
150         """Add a new recipe to the domain."""
151         name = f'recipe.{signature}'
152         anchor = f'recipe-{{signature}}'
153
154         self.data['recipe_ingredients'][name] = ingredients
155         # name, dispname, type, docname, anchor, priority
156         self.data['recipes'].append((name, signature, 'Recipe', self.
→env.docname, anchor, 0))
157
158
159     def setup(app: Sphinx) -> ExtensionMetadata:
160         app.add_domain(RecipeDomain)
161
162         return {
163             'version': '0.1',
164             'parallel_read_safe': True,
165             'parallel_write_safe': True,
166         }

```

Esaminiamo passo dopo passo ogni parte di questa estensione per spiegare cosa sta succedendo.

La classe directive

La prima cosa da esaminare è la direttiva `RecipeDirective`:

```

1  class RecipeDirective(ObjectDescription):
2      """A custom directive that describes a recipe."""
3
4      has_content = True
5      required_arguments = 1
6      option_spec = {
7          'contains': directives.unchanged_required,
8      }
9
10     def handle_signature(self, sig, signode):
11         signode += addnodes.desc_name(text=sig)
12         return sig
13
14     def add_target_and_index(self, name_cls, sig, signode):
15         signode['ids'].append('recipe' + '-' + sig)
16         if 'contains' in self.options:
17             ingredients = [x.strip() for x in self.options.get(
→'contains').split(',')]
18

```

(continues on next page)

(continua dalla pagina precedente)

```

19 recipes = self.env.get_domain('recipe')
20 recipes.add_recipe(sig, ingredients)

```

A differenza di *Estendere la sintassi con ruoli e direttive* e di *Estendere il processo di build*, questa direttiva non deriva da `docutils.parsers.rst.Directive` e non definisce un metodo `run`. Deriva invece da `sphinx.directives.ObjectDescription` e definisce i metodi `handle_signature` e `add_target_and_index`. Questo perché `ObjectDescription` è una direttiva per scopi speciali intesa a descrivere cose come classi, funzioni o, nel nostro caso, ricette. Più specificamente, `handle_signature` implementa l'analisi della firma della direttiva e passa il nome e il tipo dell'oggetto alla sua superclasse, mentre `add_target_and_index` aggiunge un target (a cui collegarsi) e una voce all'indice per questo nodo.

Vediamo anche che questa direttiva definisce `has_content`, `required_arguments` e `option_spec`. A differenza della direttiva `TodoDirective` aggiunta nel *precedente tutorial*, questa direttiva accetta un singolo argomento, il nome della ricetta e un'opzione, `contains`, oltre al `reStructuredText` annidato nel corpo.

Le classi dell'indice

Da fare: Aggiungere una breve panoramica degli indici

```

1 class IngredientIndex(Index):
2     """A custom index that creates an ingredient matrix."""
3
4     name = 'ingredient'
5     localname = 'Ingredient Index'
6     shortname = 'Ingredient'
7
8     def generate(self, docnames=None):
9         content = defaultdict(list)
10
11         recipes = {
12             name: (dispname, typ, docname, anchor)
13             for name, dispname, typ, docname, anchor, _ in self.domain.
14             ↪get_objects()
15         }
16         recipe_ingredients = self.domain.data['recipe_ingredients']
17         ingredient_recipes = defaultdict(list)
18
19         # flip from recipe_ingredients to ingredient_recipes
20         for recipe_name, ingredients in recipe_ingredients.items():
21             for ingredient in ingredients:
22                 ingredient_recipes[ingredient].append(recipe_name)

```

(continues on next page)

(continua dalla pagina precedente)

```

23         # convert the mapping of ingredient to recipes to produce the
→expected
24         # output, shown below, using the ingredient name as a key to
→group
25         #
26         # name, subtype, docname, anchor, extra, qualifier, description
27         for ingredient, recipe_names in ingredient_recipes.items():
28             for recipe_name in recipe_names:
29                 dispname, typ, docname, anchor = recipes[recipe_name]
30                 content[ingredient].append((dispname, 0, docname,
→anchor, docname, '', typ))
31
32         # convert the dict to the sorted list of tuples expected
33         content = sorted(content.items())
34
35         return content, True

```

```

1 class RecipeIndex(Index):
2     """A custom index that creates an recipe matrix."""
3
4     name = 'recipe'
5     localname = 'Recipe Index'
6     shortname = 'Recipe'
7
8     def generate(self, docnames=None):
9         content = defaultdict(list)
10
11         # sort the list of recipes in alphabetical order
12         recipes = self.domain.get_objects()
13         recipes = sorted(recipes, key=lambda recipe: recipe[0])
14
15         # generate the expected output, shown below, from the above
→using the
16         # first letter of the recipe as a key to group thing
17         #
18         # name, subtype, docname, anchor, extra, qualifier, description
19         for _name, dispname, typ, docname, anchor, _priority in
→recipes:
20             content[dispname[0].lower()].append((
21                 dispname,
22                 0,
23                 docname,
24                 anchor,
25                 docname,
26                 '',
27                 typ,

```

(continues on next page)

(continua dalla pagina precedente)

```

28         ))
29
30     # convert the dict to the sorted list of tuples expected
31     content = sorted(content.items())
32
33     return content, True

```

Sia `IngredientIndex` che `RecipeIndex` derivano da `Index`. Implementano la logica personalizzata per generare una tupla di valori che definiscono l'indice. Notare che `RecipeIndex` è un indice semplice che ha una sola voce. Estenderlo per coprire più tipi di oggetti non fa ancora parte del codice.

Entrambi gli indici utilizzano il metodo `Index.generate()` per svolgere il proprio lavoro. Questo metodo combina le informazioni del nostro dominio, le ordina e le restituisce in una struttura di una lista che sarà accettata da Sphinx. Potrebbe sembrare complicato, ma in realtà è solo un elenco di tuple come ('tomato', 'TomatoSoup', 'test', 'rec-TomatoSoup', ...). Fare riferimento alla *guida API del dominio* per ulteriori informazioni su questa API.

È possibile fare riferimento a queste pagine di indice con il ruolo `ref` combinando il nome del dominio e il valore dell'indice name. Per esempio, è possibile fare riferimento a `RecipeIndex` con `:ref:`recipe-recipe`` e `IngredientIndex` con `:ref:`recipe-ingredient``.

Il dominio

Un dominio Sphinx è un contenitore specializzato che collega insieme ruoli, direttive e indici, tra le altre cose. Diamo un'occhiata al dominio che stiamo creando qui.

```

1 class RecipeDomain(Domain):
2     name = 'recipe'
3     label = 'Recipe Sample'
4     roles = {
5         'ref': XRefRole(),
6     }
7     directives = {
8         'recipe': RecipeDirective,
9     }
10    indices = {
11        RecipeIndex,
12        IngredientIndex,
13    }
14    initial_data = {
15        'recipes': [], # object list
16        'recipe_ingredients': {}, # name -> object
17    }
18    data_version = 0
19
20    def get_full_qualified_name(self, node):

```

(continues on next page)

(continua dalla pagina precedente)

```

21     return f'recipe.{node.arguments[0]}'
22
23     def get_objects(self):
24         yield from self.data['recipes']
25
26     def resolve_xref(self, env, fromdocname, builder, typ, target,
27 → node, contnode):
28         match = [
29             (docname, anchor)
30             for name, sig, typ, docname, anchor, prio in self.get_
31 → objects()
32             if sig == target
33         ]
34
35         if len(match) > 0:
36             todocname = match[0][0]
37             targ = match[0][1]
38
39             return make_refnode(builder, fromdocname, todocname, targ,
40 → contnode, targ)
41         else:
42             print('Awww, found nothing')
43             return None
44
45     def add_recipe(self, signature, ingredients):
46         """Add a new recipe to the domain."""
47         name = f'recipe.{signature}'
48         anchor = f'recipe-{signature}'
49
50         self.data['recipe_ingredients'][name] = ingredients
51         # name, dispname, type, docname, anchor, priority
52         self.data['recipes'].append((name, signature, 'Recipe', self.
53 → env.docname, anchor, 0))

```

Ci sono alcune cose interessanti da notare su questo dominio ricetta e sui domini in generale. Innanzitutto, registriamo qui le nostre direttive, ruoli e indici, tramite gli attributi `directives`, `roles` e `indices`, piuttosto che tramite chiamate successive in `setup`. Possiamo anche notare che in realtà non stiamo definendo un ruolo personalizzato e stiamo invece riutilizzando il ruolo `sphinx.roles.XRefRole` e definendo il metodo `sphinx.domains.Domain.resolve_xref`. Questo metodo accetta due argomenti, `typ` e `target`, che si riferiscono al tipo di riferimento incrociato e al nome del target. Utilizzeremo `target` per risolvere la nostra destinazione dalle ricette del nostro dominio perché attualmente abbiamo solo un tipo di nodo.

Andando avanti, possiamo vedere che abbiamo definito `initial_data`. I valori definiti in `initial_data` verranno copiati in `env.domaindata[domain_name]` come dati iniziali del dominio e le istanze del dominio potranno accedervi tramite `self.data`. Vediamo che ab-

biamo definito due elementi in `initial_data`: `recipes` e `recipe_ingredients`. Ciascuno contiene un elenco di tutti gli oggetti definiti (ovvero tutte le ricette) e un hash che associa il nome di un ingrediente canonico all'elenco di oggetti. Il modo in cui diamo nomi agli oggetti è comune in tutta la nostra estensione ed è definito nel metodo `get_full_qualified_name`. Per ogni oggetto creato, il nome canonico è `recipe.<recipename>`, dove `<recipename>` è il nome che l'autore della documentazione dà all'oggetto (una ricetta). Ciò consente all'estensione di utilizzare diversi tipi di oggetti che condividono lo stesso nome. Avere un nome canonico e un posto centrale per i nostri oggetti è un enorme vantaggio. Sia i nostri indici che il nostro codice di riferimento incrociato utilizzano questa funzionalità.

La funzione setup

Come sempre, la funzione `setup` è un requisito e viene utilizzata per agganciare le varie parti dell'estensione a Sphinx. Diamo un'occhiata alla funzione `setup` per questa estensione.

```
1 def setup(app: Sphinx) -> ExtensionMetadata:
2     app.add_domain(RecipeDomain)
3
4     return {
5         'version': '0.1',
6         'parallel_read_safe': True,
7         'parallel_write_safe': True,
8     }
```

Sembra un po' diverso da quello che siamo abituati a vedere. Non ci sono chiamate a `add_directive()` e nemmeno a `add_role()`. Invece, abbiamo una singola chiamata a `add_domain()` seguita da un'inizializzazione del *dominio standard*. Questo perché avevamo già registrato le nostre direttive, ruoli e indici come parte della direttiva stessa.

Uso dell'estensione

Ora si può utilizzare l'estensione in tutto il progetto. Per esempio:

Listato 4: `index.rst`

Joe's Recipes

=====

Below are a collection of my favourite recipes. I highly recommend the `:recipe:ref:`TomatoSoup`` recipe in particular!

.. toctree::

tomato-soup

Listato 5: tomato-soup.rst

```
The recipe contains `tomato` and `cilantro`.
```

```
.. recipe:recipe:: TomatoSoup
   :contains: tomato, cilantro, salt, pepper
```

```
This recipe is a tasty tomato soup, combine all ingredients
and cook.
```

Le cose importanti da notare sono l'uso del ruolo `:recipe:ref:` per fare riferimenti incrociati alla ricetta effettivamente definita altrove (usando la direttiva `:recipe:recipe:`).

Ulteriori letture

Per ulteriori informazioni, fare riferimento alla documentazione [docutils](#)⁴⁰¹ e *Le API di Sphinx*.

Per condividere la propria estensione su più progetti o con altri, consultare la sezione *Estensioni di terze parti*.

Sviluppo di estensioni autodoc

L'obiettivo di questo tutorial è creare un'estensione che aggiunga il supporto per il nuovo tipo per autodoc. Questa estensione autodoc formatterà la classe `IntEnum` dalla libreria standard Python. (modulo `enum`)

Panoramica

Vogliamo che l'estensione crei la documentazione automatica per `IntEnum`. `IntEnum` è la classe `enum` intera del modulo `enum`.

Attualmente questa classe non ha un comportamento speciale di documentazione automatica.

Vogliamo aggiungere quanto segue ad autodoc:

- Una nuova direttiva `autointenum` che documenterà la classe `IntEnum`.
- La documentazione generata avrà tutti i possibili valori enumerati con i nomi.
- La direttiva `autointenum` avrà un'opzione `:hex:` che farà sì che gli interi vengano stampati in formato esadecimale.

⁴⁰¹ <https://docutils.sourceforge.io/docs/>

Prerequisiti

Abbiamo bisogno della stessa configurazione *delle estensioni precedenti*. Questa volta inseriremo l'estensione in un file chiamato `autodoc_intenum.py`. Il `my_enums.py` conterrà le enumerazioni di esempio che documenteremo.

Ecco un esempio della struttura delle cartelle che si potrebbe ottenere:

```
├── source
│   ├── _ext
│   │   └── autodoc_intenum.py
│   ├── conf.py
│   ├── index.rst
│   └── my_enums.py
```

Scrivere l'estensione

Si inizia con la funzione `setup` per l'estensione.

```
1 def setup(app: Sphinx) -> ExtensionMetadata:
2     app.setup_extension('sphinx.ext.autodoc') # Require autodoc_
   ↪ extension
3     app.add_autodocumenter(IntEnumDocumenter)
4     return {
5         'version': '1',
6         'parallel_read_safe': True,
7     }
```

Il metodo `setup_extension()` estrarrà l'estensione `autodoc` perché la nostra nuova estensione dipende da `autodoc`. `add_autodocumenter()` è il metodo che registra la nostra nuova classe di `autodocumenter`.

Vogliamo importare determinati oggetti dall'estensione `autodoc`:

```
1 from __future__ import annotations
2
3 from enum import IntEnum
4 from typing import TYPE_CHECKING, Any
5
6 from sphinx.ext.autodoc import ClassDocumenter, bool_option
7
```

Ci sono diverse classi `documenter` come `MethodDocumenter` o `AttributeDocumenter` disponibili nell'estensione `autodoc`, ma la nostra nuova classe è la sottoclasse di `ClassDocumenter` che è una classe `documenter` utilizzata da `autodoc` per documentare le classi.

Questa è la definizione della nostra nuova classe `auto-documenter`:

```

1 class IntEnumDocumenter(ClassDocumenter):
2     objtype = 'intenum'
3     directivetype = ClassDocumenter.objtype
4     priority = 10 + ClassDocumenter.priority
5     option_spec = dict(ClassDocumenter.option_spec)
6     option_spec['hex'] = bool_option
7
8     @classmethod
9     def can_document_member(
10         cls, member: Any, membername: str, isattr: bool, parent: Any
11     ) -> bool:
12         try:
13             return issubclass(member, IntEnum)
14         except TypeError:
15             return False
16
17     def add_directive_header(self, sig: str) -> None:
18         super().add_directive_header(sig)
19         self.add_line('    :final:', self.get_sourcename())
20
21     def add_content(
22         self,
23         more_content: StringList | None,
24         no_docstring: bool = False,
25     ) -> None:
26         super().add_content(more_content, no_docstring)
27
28         source_name = self.get_sourcename()
29         enum_object: IntEnum = self.object
30         use_hex = self.options.hex
31         self.add_line('', source_name)
32
33         for the_member_name, enum_member in enum_object.__members__
34         ↪ items():
35             the_member_value = enum_member.value
36             if use_hex:
37                 the_member_value = hex(the_member_value)
38
39             self.add_line(f'**{the_member_name}**: {the_member_value}',
40 ↪ source_name)
41             self.add_line('', source_name)

```

Attributi importanti della nuova classe:

objtype

Questo attributo determina il nome della direttiva auto. In questo caso la direttiva auto sarà `autointenum`.

directivetype

Questo attributo imposta il nome della direttiva generata. In questo esempio la direttiva generata sarà `.. :py:class::`.

priority

quanto più grande è il numero, tanto più alta è la priorità. Vogliamo che il nostro documenter abbia una priorità più alta rispetto al genitore.

option_spec

specifiche dell'opzione. Copiamo le opzioni della classe genitore e aggiungiamo una nuova opzione *hex*.

Membri overridden [*sovrascritti*]:

can_document_member

È importante sovrascrivere questo membro. Dovrebbe restituire *True* quando l'oggetto passato può essere documentato da questa classe.

add_directive_header

Questo metodo genera l'header [*intestazione*] della direttiva. Aggiungiamo l'opzione direttiva **:final:**. Ricordarsi di chiamare **super** altrimenti non verrà generata alcuna direttiva.

add_content

Questo metodo genera il corpo della documentazione della classe. Dopo aver chiamato il metodo `super` generiamo le righe per la descrizione dell'enum.

Uso dell'estensione

Ora si può utilizzare la nuova direttiva autodoc per documentare qualsiasi `IntEnum`.

Ad esempio, avendo il seguente `IntEnum`:

Listato 6: `my_enums.py`

```
class Colors(IntEnum):
    """Colors enumerator"""
    NONE = 0
    RED = 1
    GREEN = 2
    BLUE = 3
```

Questo sarà il file di documentazione con la direttiva di auto-documentazione:

Listato 7: index.rst

```
.. autointenum:: my_enums.Colors
```

Ulteriori letture

Per condividere la propria estensione su più progetti o con altri, consultare la sezione [Estensioni di terze parti](#).

5.2 Gli How-to

Dipendenza da un'altra estensione

A volte la propria estensione dipende dalla funzionalità di un'altra estensione Sphinx. La maggior parte delle estensioni Sphinx vengono attivate nel file `conf.py` di un progetto, ma questo non è disponibile allo sviluppatore di estensioni.

Per garantire che un'altra estensione venga attivata come parte della propria estensione, si utilizza il metodo `sphinx.application.Sphinx.setup_extension()`. Ciò attiverà un'altra estensione in fase di esecuzione, assicurando l'accesso alle sue funzionalità.

Ad esempio, il codice seguente attiva l'estensione `sphinx.ext.autodoc`:

```
def setup(app):  
    app.setup_extension('sphinx.ext.autodoc')
```

Nota: Poiché la propria estensione dipenderà da un'altra, ci si assicuri di includerla come parte dei requisiti di installazione dell'estensione.

Configurazione dei builder

Evidenziare l'entry point dei builder

Added in version 1.6.

Le estensioni *builder* possono essere scoperte tramite gli *entry points*⁴⁰² in modo che non debbano essere elencati nel valore di configurazione *extensions*.

Le estensioni del builder dovrebbero definire un punto di ingresso nel gruppo "sphinx.builders". Il nome del punto di ingresso deve corrispondere all'attributo *name* del builder, che è il nome passato all'opzione `sphinx-build -b`. Il valore del punto di ingresso dovrebbe

⁴⁰² https://setuptools.pypa.io/en/latest/userguide/entry_point.html

essere uguale al nome puntato [dotted] del modulo dell'estensione. Ecco un esempio di come è possibile definire un punto di ingresso per 'mybuilder' nel file `pyproject.toml` dell'estensione

```
[project.entry-points."sphinx.builders"]
mybuilder = "my.extension.module"
```

Da notare che è ancora necessario registrare il builder utilizzando `add_builder()` nella funzione `setup()` dell'estensione.

5.3 Sviluppo di temi HTML

Added in version 0.6.

Nota: Questo documento fornisce informazioni sulla creazione del proprio tema. Per utilizzare temi HTML preesistenti, fare riferimento a *Temi HTML*.

Sphinx supporta la modifica dell'aspetto del suo output HTML tramite i *temi*. Un tema è una raccolta di template HTML, fogli di stile e altri file statici. Inoltre, ha un file di configurazione che specifica da quale tema ereditare, quale stile di evidenziazione utilizzare e quali opzioni esistono per personalizzare l'aspetto del tema.

I temi sono pensati per essere inconsapevoli del progetto, quindi possono essere utilizzati per progetti diversi senza modifiche.

Nota: Consultare *Le API di Sphinx* per maggiori informazioni che potrebbero essere utili nello sviluppo dei temi.

Creazione di temi

I temi assumono la forma di una directory o di un file zip (il cui nome è quello del tema), contenente quanto segue:

- Un file `theme.toml` (preferito) o un file `theme.conf`.
- Template HTML, se necessari.
- Una directory `static/` contenente tutti i file statici che verranno copiati nella directory statica di output durante build. Possono essere immagini, stili, file di script.

Configurazione del tema (`theme.toml`)

Il file `theme.toml` è un documento TOML⁴⁰³, contenente due tabelle: `[theme]` e `[options]`.

La tabella `[theme]` definisce le impostazioni del tema:

- **`inherit`** (stringa): Il nome del tema di base da cui ereditare impostazioni, opzioni, template e file statici. Verranno utilizzati tutti i file statici dei temi 'antenati'. Il tema utilizzerà tutte le opzioni definite nei temi ereditati `[inherited]`. Infine, i temi ereditati `[inherited]` verranno utilizzati per individuare i template mancanti (ad esempio, se come tema di base viene utilizzato `"basic"`, la maggior parte dei modelli sarà già definita).

Se impostato su `"none"`, il tema non erediterà da nessun altro tema. L'ereditarietà è ricorsiva e forma una catena di temi ereditati (ad esempio `default -> classic -> basic -> none`).

- **`stylesheets`** (elenco di stringhe): Un elenco di nomi di file CSS che verranno inclusi nell'header HTML generato. L'impostazione del valore di configurazione `html_style` sovrascriverà `[override]` questa impostazione.

Altri meccanismi per includere più fogli di stile includono `@import` nei CSS o l'utilizzo di un template HTML personalizzato con tag `<link rel="stylesheet">` appropriati.

- **`sidebars`** (elenco di stringhe): Un elenco di template di barre laterali. Questo può essere sovrascritto dall'utente tramite il valore di configurazione `html_sidebars`.
- **`pygments_style`** (tabella): Una tabella TOML che definisce i nomi degli stili Pygments da utilizzare per evidenziare la sintassi. La tabella ha due chiavi `[key]` riconosciute: `default` e `dark`. Lo stile definito nella chiave `dark` verrà utilizzato quando la query multimediale CSS (`prefers-color-scheme: dark`) restituisce `true`.

`[theme.pygments_style.default]` può essere sovrascritto dall'utente tramite il valore di configurazione `pygments_style`.

La tabella `[options]` definisce le opzioni per il tema. È strutturata in modo tale che ciascuna coppia chiave-valore corrisponda a un nome di variabile e al corrispondente valore di default. Queste opzioni possono essere sovrascritte dall'utente in `html_theme_options` e sono accessibili da tutti i template come `theme_<name>`.

Added in version 7.3: supporto per `theme.toml`.

File `theme.toml` di esempio:

```
[theme]
inherit = "basic"
stylesheets = [
    "main-CSS-stylesheet.css",
]
sidebars = [
    "localtoc.html",
    "relations.html",
```

(continues on next page)

⁴⁰³ <https://toml.io/en/>

(continua dalla pagina precedente)

```

    "sourcelink.html",
    "searchbox.html",
]
# Style names from https://pygments.org/styles/
pygments_style = { default = "style_name", dark = "dark_style" }

[options]
variable = "default value"

```

Configurazione del tema (theme.conf)

Il file `theme.conf` è in formato INI¹ (leggibile dal modulo standard Python `configparser`⁴⁰⁴) e ha la seguente struttura:

```

[theme]
inherit = base theme
stylesheet = main CSS name
pygments_style = stylename
sidebars = localtoc.html, relations.html, sourcelink.html, searchbox.
→html

[options]
variable = default value

```

- L'impostazione di **inherit** dà il nome di un "base theme", o `none`. Il "base theme" verrà utilizzato per individuare i template mancanti (la maggior parte dei temi non dovrà fornire la maggior parte dei template se utilizzano `basic` con "base theme"), le sue opzioni verranno ereditate e verranno anche utilizzati tutti i suoi file statici. Per ereditare anche il foglio di stile, lo si include nel proprio tramite l'`@import` del CSS.
- L'impostazione **stylesheet** fornisce un elenco di nomi di file CSS separati da virgole a cui verrà fatto riferimento nell'header HTML. La tecnica dell'`@import` CSS è utilizzabile per includere l'uno dall'altro, oppure si può utilizzare un template HTML personalizzato che aggiunga i tag `<link rel="stylesheet">` necessari. L'impostazione del valore di configurazione `html_style` sovrascriverà questa impostazione.
- L'impostazione **pygments_style** fornisce il nome di uno stile Pygments da utilizzare per l'evidenziazione. Questo può essere sovrascritto dall'utente nel valore di configurazione `pygments_style`.
- L'impostazione **pygments_dark_style** fornisce il nome di uno stile Pygments da utilizzare per l'evidenziazione quando la query multimediale CSS (`prefers-color-scheme: dark`) restituisce `true`. Viene inserito nella pagina utilizzando `add_css_file()`.

¹ Non è un file Python eseguibile, al contrario di `conf.py`, perché ciò rappresenterebbe un rischio non necessario per la sicurezza se i temi fossero condivisi.

⁴⁰⁴ <https://docs.python.org/3/library/configparser.html#module-configparser>

- L'impostazione **sidebars** fornisce l'elenco separato da virgole di modelli di barre laterali per la costruzione di barre laterali. Questo può essere sovrascritto dall'utente nel valore di configurazione `html_sidebars`.
- La sezione **options** contiene coppie di nomi di variabili e valori di default. Queste opzioni possono essere sovrascritte dall'utente in `html_theme_options` e sono accessibili da tutti i template come `theme_<name>`.

Added in version 1.7: impostazioni della sidebar

Cambiato nella versione 5.1: L'impostazione del foglio di stile accetta più filename CSS

Convertire `theme.conf` in `theme.toml`

I file di configurazione del tema in stile INI (`theme.conf`) possono essere convertiti in TOML tramite un programma di supporto distribuito con Sphinx. È previsto per un utilizzo una tantum e potrebbe essere rimosso senza preavviso in una versione futura di Sphinx.

```
$ python -m sphinx.theming conf_to_toml [THEME DIRECTORY PATH]
```

L'argomento richiesto è il percorso di una directory contenente un file `theme.conf`. Il programma scriverà un file `theme.toml` nella stessa directory e non modificherà il file `theme.conf` originale.

Added in version 7.3.

Distribuire il tema come pacchetto Python

Come modalità per distribuire il tema, si può utilizzare un pacchetto Python. Ciò semplifica la configurazione del tema da parte degli utenti.

Per distribuire il tema come pacchetto Python, si definisce un punto di ingresso chiamato `sphinx.html_themes` nel file `pyproject.toml` e si scrive una funzione `setup()` per registrare il tema utilizzando il comando `add_html_theme()` dell'API:

```
# pyproject.toml

[project.entry-points."sphinx.html_themes"]
name_of_theme = "your_theme_package"
```

```
# your_theme_package.py
from os import path

def setup(app):
    app.add_html_theme('name_of_theme', path.abspath(path.dirname(__
    ↪ file__)))
```

Se il pacchetto dei temi contiene due o più temi, si chiama `add_html_theme()` due o più volte.

Added in version 1.2: funzionalità entry_points 'sphinx_themes'.

Deprecato dalla versione 1.6: l'entry_points sphinx_themes è stato deprecato.

Added in version 1.6: funzionalità entry_points sphinx.html_themes.

I template

I template

Sphinx utilizza il motore di template [Jinja](https://jinja.palletsprojects.com/)⁴⁰⁵ per i suoi template HTML. Jinja è un motore text-based (*basato su testo*), ispirato ai template Django, quindi chiunque abbia utilizzato Django ne avrà già familiarità. Ha anche un'eccellente documentazione per coloro che hanno bisogno di familiarizzarsi con esso.

Si devono utilizzare i template di Sphinx per produrre HTML?

No. Ci sono diverse opzioni:

- Si può scrivere una sottoclasse `TemplateBridge` che chiami il motore di template preferito e impostare il valore di configurazione `template_bridge` di conseguenza.
- Si può *scrivere un builder personalizzato* che derivi da `StandaloneHTMLBuilder` e chiamare il motore di template scelto.
- Si può usare `PickleHTMLBuilder` che produce file pickle con il contenuto della pagina e post-elaborarli utilizzando uno strumento personalizzato o utilizzarli nell'applicazione Web.

Elementi sui Template Jinja/Sphinx

Il linguaggio di default dei template in Sphinx è Jinja. È ispirato a Django/Smarty ed è facile da capire. Il concetto più importante in Jinja è l'*ereditarietà del template*, il che significa che si possono sovrascrivere solo blocchi specifici all'interno di un template, personalizzandolo e mantenendo le modifiche al minimo.

Per personalizzare l'output della documentazione si possono sovrascrivere tutti i template (sia i quelli di layout che i template figli) aggiungendo file con lo stesso nome del file originale nella directory dei template della struttura generata per dall quickstart di Sphinx.

Sphinx cercherà prima i template nelle cartelle di `templates_path` e, se non riesce a trovare lì il template che sta cercando, ricorre ai template del tema selezionato.

Un template contiene **variabili**, che vengono sostituite con valori quando il modello viene valutato, **tag**, che controllano la logica del template e **blocchi** utilizzati per l'ereditarietà del template.

Il tema *basic* di Sphinx fornisce i template di base con un paio di blocchi che riempirà con dei dati. Questi si trovano nella sottodirectory `themes/basic` della directory di installazione

⁴⁰⁵ <https://jinja.palletsprojects.com/>

di Sphinx e sono utilizzati da tutti i temi nativi di Sphinx. I template con lo stesso nome in `templates_path` sovrascrivono quelli forniti dal tema selezionato.

Ad esempio, per aggiungere un nuovo link all'area del template contenente i link correlati, tutto ciò che si deve fare è aggiungere un nuovo template chiamato `layout.html` con i seguenti contenuti:

```
{% extends "!layout.html" %}
{% block rootrellink %}
    <li><a href="https://project.invalid/">Project Homepage</a> &raquo;
    </li>
    {{ super() }}
{% endblock %}
```

Anteponendo un punto esclamativo al nome del template overridden [sovrascritto], Sphinx caricherà il template di layout dal tema HTML sottostante.

Importante: Se si sovrascrive un blocco, si chiama `{{ super() }}` da qualche parte per visualizzare il contenuto originale del blocco nel template esteso, a meno che non si voglia che quel contenuto venga visualizzato.

Lavorare con i template nativi

Il tema **basic** nativo fornisce i template su cui si basano tutti i temi Sphinx nativi. Ha i seguenti elementi che si possono sovrascrivere o utilizzare:

Blocchi

Nel template `layout.html` esistono i seguenti blocchi:

doctype

Il doctype del formato di output. Per default è XHTML 1.0 Transitional poiché è il più vicino a ciò che Sphinx e Docutils generano ed è una buona idea non cambiarlo a meno che non si voglia passare a HTML 5 o ad un doctype XHTML diverso ma compatibile.

linktags

Questo blocco aggiunge un paio di tag `<link>` alla sezione head del template.

extrahead

Questo blocco è vuoto per default ed è utilizzabile per aggiungere contenuti extra nel tag `<head>` del file HTML generato. Questo è il posto giusto per aggiungere riferimenti a JavaScript o a file CSS aggiuntivi.

relbar1, relbar2

Questo blocco contiene la *barra delle relazioni*, l'elenco dei link correlati (i documenti principali a sinistra e i link all'indice, ai moduli ecc. a destra). `relbar1` appare prima del documento, `relbar2` dopo. Per default, entrambi i blocchi vengono riempiti; per

mostrare la relbar solo prima del documento, si deve sovrascrivere `relbar2` in questo modo:

```
{% block relbar2 %}{% endblock %}
```

rootrellink, relbaritems

All'interno della relbar ci sono tre sezioni: Il `rootrellink`, i link dalla documentazione e i `relbaritems` personalizzati. Il `rootrellink` è un blocco che per default contiene un elemento dell'elenco che punta al documento root, mentre il `relbaritems` è un blocco vuoto. Se si sovrascrivono per aggiungere collegamenti extra nella barra, devono essere elementi dell'elenco e devono terminare con `reldelim1`.

document

Il contenuto del documento stesso. Contiene il blocco "body" in cui il contenuto individuale viene inserito da sub-template come `page.html`.

Nota: Affinché la ricerca JavaScript nativa mostri un'anteprima della pagina nella pagina dei risultati, il documento o il contenuto del corpo devono essere racchiuso in un elemento HTML contenente l'attributo `role="main"`. Per esempio:

```
<div role="main">
  {% block document %}{% endblock %}
</div>
```

sidebar1, sidebar2

Una possibile posizione per una barra laterale. `sidebar1` appare prima del documento ed è vuoto per default, `sidebar2` dopo il documento e contiene la barra laterale di default. Per scambiare la posizione della barra laterale, la si sovrascrive e si chiama l'helper `sidebar`:

```
{% block sidebar1 %}{{ sidebar() }}{% endblock %}
{% block sidebar2 %}{% endblock %}
```

(La posizione `sidebar2` per la barra laterale è necessaria, ad esempio, per il foglio di stile `sphinxdoc.css`.)

sidebarlogo

La posizione del logo all'interno della barra laterale. Lo si sovrascrive per posizionare alcuni contenuti nella parte superiore della barra laterale.

footer

Il blocco per il div footer. Per avere un piè di pagina o un markup personalizzato prima o dopo, sovrascrivere questo.

I seguenti quattro blocchi sono usati *solo* per le pagine a cui non è stato assegnato un elenco di barre laterali personalizzate nel valore di configurazione `html_sidebars`. Il loro utilizzo è deprecato a favore di modelli di barre laterali separati, che possono essere inclusi tramite `html_sidebars`.

sidebartoc

Il sommario nella barra laterale.

Deprecato dalla versione 1.0.

sidebarrel

La i link correlati (documento precedente, successivo) all'interno della barra laterale.

Deprecato dalla versione 1.0.

sidebarsourcelink

Il link "Mostra sorgente" nella barra laterale (normalmente mostrato solo se è abilitato da [`html_show_sourcelink`](#)).

Deprecato dalla versione 1.0.

sidebarsearch

La casella di ricerca nella barra laterale. Lo si sovrascrive per posizionare del contenuto nella parte inferiore della barra laterale.

Deprecato dalla versione 1.0.

Variabili di Configurazione

All'interno dei template si possono impostare un paio di variabili utilizzate dal template di layout utilizzando il tag `{% set %}`:

reldelim1

Il delimitatore per gli elementi sul lato sinistro della barra correlata. Il default è ' »'. Ogni elemento nella barra correlata termina con il valore di questa variabile.

reldelim2

Il delimitatore per gli elementi sul lato destro della barra correlata. Il default è ' |'. Ogni voce tranne l'ultima della relativa barra termina con il valore di questa variabile.

L'override funziona in questo modo:

```
{% extends "!layout.html" %}
{% set reldelim1 = ' &gt;' %}
```

script_files

Aggiunge qui ulteriori file di script, in questo modo:

```
{% set script_files = script_files + ["_static/myscript.js"] %}
```

Deprecato dalla versione 1.8.0: Utilizzare invece `.Sphinx.add_js_file()`.

Funzioni Helper

Sphinx fornisce varie funzioni Jinja come helper nel template. Si possono usare per generare link o generare elementi utilizzati più volte.

pathto(*document*)

Restituisce il percorso di un documento Sphinx come URL. Lo si usa per fare riferimento a documenti *build-ati*.

pathto(*file*, 1)

Restituisce il path di un *file* che è un filename relativo alla radice dell'output generato. Lo si usa per fare riferimento a file statici.

hasdoc(*document*)

Controlla se esiste un documento con il nome *document*.

sidebar()

Restituisce la barra laterale visualizzata.

relbar()

Restituisce la barra della relazione visualizzata.

warning(*message*)

Emette un messaggio di warning.

Variabili Globali

Queste variabili globali sono disponibili in ogni template e sono sicure da usare. Ce ne sono altre, ma la maggior parte di esse rappresenta un dettaglio di implementazione e potrebbero cambiare in futuro.

builder

Il nome del builder (ad esempio `html` or `htmlhelp`).

copyright

Il valore di *copyright*.

docstitle

Il titolo della documentazione (il valore di *html_title*), tranne quando viene utilizzato il builder "single-file", quando è impostato su `None`.

embedded

True se l'HTML creato è pensato per essere incluso in qualche applicazione di visualizzazione che gestisce la navigazione, non nel browser web, come per la guida HTML o i formati help di Qt. In questo caso, la barra laterale non è inclusa.

favicon_url

Il path relativo all'immagine della favicon HTML dal documento corrente, o l'URL della favicon, o ''.

Added in version 4.0.

file_suffix

Il valore dell'attributo *out_suffix* del builder, ovvero l'estensione del nome file che riceveranno i file di output. Per un builder HTML standard, solitamente è `.html`.

has_source

True se i sorgenti del documento reST vengono copiati (se *html_copy_source* è True).

language

Il valore di *language*.

last_updated

La data della build.

logo_url

Il path relativo all'immagine del logo HTML dal documento corrente, o l'URL al logo, o `''`.

Added in version 4.0.

master_doc

Lo stesso di *root_doc*.

Cambiato nella versione 4.0: Rinominato in *root_doc*.

root_doc

Il valore di *root_doc*, da utilizzare con *path_to()*.

Cambiato nella versione 4.0: Rinominato da *master_doc*.

pagename

Il "page name" del file corrente, ovvero il nome del documento se il file è generato da una sorgente reST, o il nome gerarchico equivalente relativo alla directory di output (`[directory/]filename_without_extension`).

project

Il valore di *project*.

release

Il valore di *release*.

rellinks

Un elenco di link da inserire sul lato sinistro della relbar, accanto a "next" e "prev". Questo di solito contiene link all'indice generale e ad altri indici, come l'indice del modulo Python. Se si aggiunge qualcosa, deve essere una tupla (nome pagina, titolo link, chiave di accesso, testo del link).

shorttitle

Il valore di *html_short_title*.

show_source

True se *html_show_sourcelink* è True.

sphinx_version

La versione di Sphinx utilizzata per la build è rappresentata come una stringa, ad esempio "3.5.1".

sphinx_version_tuple

La versione di Sphinx utilizzata per la build è rappresentata come una tupla di cinque elementi. Per Sphinx versione 3.5.1 beta 3 questo sarebbe (3, 5, 1, 'beta', 3). Il quarto elemento può essere uno tra: `alpha`, `beta`, `rc`, `final`. `final` ha sempre 0 come ultimo elemento.

Added in version 4.2.

docutils_version_info

La versione di Docutils utilizzata per creare è rappresentata come una tupla di cinque elementi. Per Docutils versione 0.16.1 beta 2 questo sarebbe (0, 16, 1, 'beta', 2). Il quarto elemento può essere uno tra: `alpha`, `beta`, `candidate`, `final`. `final` ha sempre 0 come ultimo elemento.

Added in version 5.0.2.

styles

Un elenco dei nomi dei fogli di stile principali forniti dal tema o da [html_style](#).

Added in version 5.1.

title

Il titolo del documento corrente, come utilizzato nel tag `<title>`.

use_opensearch

Il valore di [html_use_opensearch](#).

version

Il valore di [version](#).

Oltre a questi valori, sono disponibili anche tutte le **opzioni del theme** (con il prefisso `theme_`), nonché i valori forniti dall'utente in [html_context](#).

Nei documenti creati da file sorgenti (a differenza dei file generati automaticamente come l'indice del modulo o dei documenti già in formato HTML), sono disponibili anche queste variabili:

body

Una stringa contenente il contenuto della pagina in formato HTML prodotto dal builder HTML, prima dell'applicazione del tema.

display_toc

Un booleano che è True se il sommario contiene più di una voce.

meta

Metadati del documento (un dizionario), vedere [Metadati a livello di file](#).

metatags

Una stringa contenente i tag HTML [meta](#)⁴⁰⁶ della pagina.

next

Il prossimo documento per la navigazione. Questa variabile è false o ha due attributi `link` e `title`. Il titolo contiene markup HTML. Ad esempio, per generare un link alla pagina successiva, si può utilizzare questo snippet:

```
{% if next %}
<a href="{{ next.link|e }}">{{ next.title }}</a>
{% endif %}
```

page_source_suffix

Il suffisso del file di cui è stato eseguito il rendering. Poiché supportiamo un elenco di `source_suffix`, questo consentirà di collegarsi correttamente al file sorgente originale.

parents

Un elenco di documenti genitori per la navigazione, strutturato come l'elemento `next`.

prev

Come `next`, ma per la pagina precedente.

sourcename

Il nome del file sorgente copiato per il documento corrente. Questo non è vuoto solo se il valore di `html_copy_source` è True. Questo ha un valore vuoto nella creazione di file generati automaticamente.

toc

Il sommario locale della pagina corrente, visualizzato come elenchi puntati HTML.

toctree

Un *callable* che fornisce l'albero del sommario globale contenente la pagina corrente, visualizzato come elenco puntato HTML. Argomenti facoltativi delle parole chiave:

collapse

Se true, tutte le voci del sommario che non sono antenati della pagina corrente verranno compresse. True per default.

maxdepth

La profondità massima dell'albero. Lo si imposta a -1 per avere una profondità illimitata. Il default è la profondità massima selezionata nella direttiva `toctree`.

titles_only

Se true, inserisce nell'albero solo i titoli dei documenti di livello superiore. False per default.

includehidden

Se true, l'albero del ToC conterrà anche voci nascoste. False per default.

La *guida ai template* è utile per scrivere i template. Ciò che è importante tenere a mente è l'ordine in cui Sphinx cerca i template:

- Innanzitutto, nelle directory `templates_path` dell'utente.
- Poi, nel tema selezionato.

⁴⁰⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#meta>

- Quindi, nel suo tema di base, nel tema di base della sua base, ecc.

Quando si estende un template nel tema di base con lo stesso nome, si usa il nome del tema come una directory esplicita: `{% extends "basic/layout.html" %}`. Da un template utente `templates_path`, è comunque possibile utilizzare la sintassi del "punto esclamativo" come *descritto nel documento del template*.

Template statici

Poiché le opzioni del tema sono pensate per consentire facilitare all'utente la configurare, senza dover scrivere un foglio di stile personalizzato, è necessario essere in grado di creare template di file statici oltre a file HTML. Pertanto, Sphinx supporta i cosiddetti "template statici", come questo:

Se il nome di un file nella directory `static/` di un tema (o nel percorso di `static` dell'utente) termina con `.jinja` o con `_t`, verrà elaborato dal motore dei template. Il suffisso verrà rimosso dal nome del file finale.

Ad esempio, un tema con un file `static/theme_styles.css.jinja` potrebbe utilizzare i template per inserire opzioni nel foglio di stile. Quando un progetto di documentazione viene creato con quel tema, la directory di output conterrà un file `_static/theme_styles.css` in cui tutti i tag del template sono stati elaborati.

Cambiato nella versione 7.4:

Il suffisso preferito per i template statici è ora `.jinja`, in linea con [l'estensione consigliata del file⁴⁰⁷](#) del progetto Jinja.

Il suffisso del file `_t` per i template statici è ora considerato 'legacy' e il supporto potrebbe eventualmente essere rimosso.

Se viene rilevato un template statico con il suffisso `_t` o `.jinja`, verrà elaborato dal motore dei template, con il suffisso rimosso dal nome del file finale.

Uso di metadati di pagina personalizzati nei template HTML

Qualsiasi coppia chiave/valore nelle *liste di campi* posizionata *prima* del titolo della pagina sarà disponibile per il template Jinja durante la build della pagina all'interno dell'attributo `meta`. Ad esempio, se una pagina avesse il seguente testo prima del primo titolo:

```
:mykey: My value
```

```
My first title
```

```
-----
```

Quindi è possibile accedervi all'interno di un modello Jinja in questo modo:

⁴⁰⁷ <https://jinja.palletsprojects.com/en/latest/templates/#template-file-extension>

```
{%- if meta is mapping %}
    {{ meta.get("mykey") }}
{%- endif %}
```

Da notare il controllo che `meta` sia un dictionary ("mapping" nella terminologia Jinja) per garantire che l'utilizzo in questo modo sia valido.

Definizione delle funzioni del template personalizzato

A volte è utile definire la propria funzione in Python che si desidera utilizzare in un template. Ad esempio, per inserire un valore template con una logica che dipende dalla configurazione dell'utente nel progetto o per includere controlli non banali e fornire messaggi di errore semplici per una configurazione errata nel template.

Per definire la funzione template, si devono definire due funzioni all'interno del modulo:

- Una funzione **event handler del contesto della pagina** (o **registration**). Questa è connessa all'applicazione *Sphinx* tramite una callback di eventi.
- Una **funzione template** che si userà nel template Jinja.

Innanzitutto, si definisce la funzione di registrazione, che accetta gli argomenti per *html-page-context*.

Nella funzione di registrazione, si definisce la funzione del template che si desidera utilizzare all'interno di Jinja. La funzione template dovrebbe restituire una stringa o oggetti Python (liste, dizionari) con stringhe che Jinja utilizza internamente nel processo dei template

Nota: La funzione template avrà accesso a tutte le variabili passate alla funzione di registrazione.

Alla fine della funzione di registrazione, si aggiunge la funzione template al contesto dell'applicazione Sphinx con `context['template_func'] = template_func`.

Infine, nella funzione `setup()`, dell'estensione, si aggiunge la funzione di registrazione come callback per *html-page-context*.

```
# The registration function
def setup_my_func(app, pagename, templatename, context, doctree):
    # The template function
    def my_func(mystring):
        return "Your string is %s" % mystring
    # Add it to the page's context
    context['my_func'] = my_func

# Your extension's setup function
def setup(app):
    app.connect("html-page-context", setup_my_func)
```

Ora accederà a questa funzione in jinja in questo modo:

```
<div>
{{ my_func("some string") }}
</div>
```

Aggiunta dei file statici alle risorse della build

Per default, Sphinx copia i file statici nella directory `static/` della directory template. Tuttavia, se per qualche motivo il pacchetto ha bisogno di posizionare file statici al di fuori della directory `static/`, si devono copiare manualmente nella directory `_static/` degli output HTML durante la build tramite un hook di evento. Ecco un esempio di codice per ottenere questo risultato:

```
from os import path
from sphinx.util.fileutil import copy_asset_file

def copy_custom_files(app, exc):
    if app.builder.format == 'html' and not exc:
        staticdir = path.join(app.builder.outdir, '_static')
        copy_asset_file('path/to/myextension/_static/myjsfile.js',
            staticdir)

def setup(app):
    app.connect('build-finished', copy_custom_files)
```

Iniettare JavaScript in base alla configurazione dell'utente

Se l'estensione fa uso di JavaScript, può essere utile consentire agli utenti di controllarne il comportamento utilizzando la configurazione Sphinx. Tuttavia, questo può essere difficile da fare se il JavaScript si presenta sotto forma di libreria statica (che non verrà buildata con Jinja).

Esistono due modi per iniettare (inserire) variabili nello spazio JavaScript in base alla configurazione dell'utente.

Innanzitutto, si può aggiungere `_t` alla fine di qualsiasi file statico incluso nell'estensione. Ciò farà sì che Sphinx elabori questi file con il motore dei template, consentendoti di incorporare variabili e controllare il comportamento.

Ad esempio, la seguente struttura JavaScript:

```
mymodule/
├── _static
│   └── myjsfile.js_t
└── mymodule.py
```

Il risultato sarà il seguente file statico inserito nell'output di build dell'HTML:

```
_build/
├── html
│   └── _static
│       └── myjsfile.js
```

Consultare *Template statici* per maggiori informazioni.

In secondo luogo, si può utilizzare il metodo `Sphinx.add_js_file()` senza passargli un file. Normalmente, questo metodo viene utilizzato per inserire un nuovo file JavaScript nel sito. Tuttavia, se *non* gli si passa un path di file, ma si passa invece una stringa all'argomento "body", allora questo testo verrà inserito come JavaScript nell'intestazione del sito. Ciò consente di inserire variabili nel JavaScript del progetto da Python.

Ad esempio, il codice seguente leggerà un valore configurato dall'utente e poi inserirà questo valore come variabile JavaScript, che il codice JavaScript dell'estensione potrebbe utilizzare:

```
# This function reads in a variable and inserts it into JavaScript
def add_js_variable(app):
    # This is a configuration that you've specified for users in `conf.py`
    js_variable = app.config['my_javascript_variable']
    js_text = "var my_variable = '%s';" % js_variable
    app.add_js_file(None, body=js_text)
# We connect this function to the step after the builder is initialized
def setup(app):
    # Tell Sphinx about this configuration variable
    app.add_config_value('my_javascript_variable', 0, 'html')
    # Run the function after the builder is initialized
    app.connect('builder-inited', add_js_variable)
```

Di conseguenza, nel tema si può utilizzare un codice che dipende dalla presenza di questa variabile. Gli utenti possono controllare il valore della variabile definendolo nel loro file `conf.py`.

CAPITOLO 6

Le API di Sphinx

Poiché molti progetti avranno bisogno di funzionalità speciali nella loro documentazione, Sphinx è progettato per essere estensibile su più livelli.

Ecco alcune cose che si possono fare in un'estensione:

- Aggiungere nuovi *builder* per supportare nuovi formati di output o azioni sui documenti analizzati.
- Registrare ruoli e direttive reStructuredText personalizzati, estendendo il markup utilizzando *API di markup Docutils*.
- Aggiungere codice personalizzato ai cosiddetti "hook points" (*punti di aggancio*) in punti strategici durante tutto il processo di creazione, per registrare un hook ed eseguire codice specializzato. Per un esempio, vedere gli *API di callback degli eventi*.

Un'estensione è semplicemente un modulo Python con una funzione `setup()`. Un utente attiva l'estensione inserendo il nome del modulo dell'estensione (o un sottomodulo) nel proprio valore di configurazione *extensions*.

Quando viene eseguito **sphinx-build**, Sphinx tenterà di importare ogni modulo elencato ed eseguire `yourmodule.setup(app)`. Questa funzione viene utilizzata per preparare l'estensione (ad esempio, eseguendo il codice Python), collegando le risorse che Sphinx utilizza nel processo di build (come file CSS o HTML) e notificando a Sphinx tutto ciò che l'estensione offre (come direttive o definizioni di ruoli). L'argomento `app` è un'istanza di *Sphinx* e dà il controllo sulla maggior parte degli aspetti della build Sphinx.

Nota: Il file di configurazione stesso può essere trattato come un'estensione se contiene una funzione `setup()`. Tutte le altre estensioni da caricare devono essere elencate nel valore di configurazione *extensions*.

Il resto di questa pagina descrive alcuni aspetti di alto livello dello sviluppo delle estensioni e varie parti del comportamento di Sphinx che si possono controllare. Per alcuni esempi di come le estensioni possono essere create e utilizzate per controllare diverse parti di Sphinx, vedere il [Tutorial](#).

6.1 Oggetti importanti

Esistono diversi oggetti chiave di se ne utilizzeranno le API durante la scrittura di un'estensione. Questi sono:

Application

L'oggetto applicazione (solitamente chiamato `app`) è un'istanza di [Sphinx](#). Controlla la maggior parte delle funzionalità di alto livello, come l'impostazione delle estensioni, l'invio di eventi e la produzione di output (logging).

Se si dispone dell'oggetto `environment` [*ambiente*], l'applicazione è disponibile come `env.app`.

Environment

L'oggetto dell'`environment` [*ambiente*] di compilazione (solitamente chiamato `env`) è un'istanza di [BuildEnvironment](#). È responsabile dell'analisi dei documenti sorgenti, memorizza tutti i metadati sulla raccolta di documenti e viene serializzato su disco dopo ogni build.

La sua API fornisce metodi relativi all'accesso ai metadati, alla risoluzione dei riferimenti, ecc. Può anche essere utilizzata dalle estensioni per memorizzare nella cache le informazioni che dovrebbero persistere per le re-build incrementali.

Se si dispone dell'oggetto `application` o `builder`, l'ambiente è disponibile come `app.env` o `builder.env`.

Builder

L'oggetto `builder` (solitamente chiamato `builder`) è un'istanza di una sottoclasse specifica di [Builder](#). Ogni classe `builder` sa come convertire i documenti analizzati in un formato di output o elaborarli in altro modo (ad esempio controllare i link [*collegamenti*] esterni).

Se ci ha l'oggetto `application`, il `builder` è disponibile come `app.builder`.

Config

L'oggetto `config` (solitamente chiamato `config`) fornisce i valori di configurazione impostati in `conf.py` come attributi. È un'istanza di [Config](#).

La configurazione è disponibile come `app.config` o `env.config`.

Per vedere un esempio di utilizzo di questi oggetti, fare riferimento [ai tutorial](#).

6.2 Fasi della Build

Una cosa fondamentale per comprendere i meccanismi di estensione è il modo in cui viene buildato [*costruito*] un progetto Sphinx: questo funziona in più fasi.

Fase 0: Inizializzazione

In questa fase non accade quasi nulla che ci interessi. Nella directory sorgente vengono cercati i file sorgenti e le estensioni vengono inizializzate. Se esiste un ambiente di build memorizzato, viene caricato, altrimenti ne viene creato uno nuovo.

Fase 1: Lettura

Nella Fase 1, tutti i file sorgente (e nelle build successive, quelli nuovi o modificati) vengono letti e analizzati. Questa è la fase in cui le direttive e i ruoli vengono incontrati dai docutil e il codice corrispondente viene eseguito. L'output di questa fase è un *doctree* per ogni file sorgente; questo è un albero di nodi docutils. Per gli elementi del documento che non sono completamente noti finché non vengono letti tutti i file esistenti, vengono creati dei nodi temporanei.

Ci sono nodi forniti da docutils, che sono documentati [nella documentazione di docutils](#)⁴⁰⁸. Nodi aggiuntivi sono forniti da Sphinx e *documentati qui*.

Durante la lettura, l'ambiente di compilazione viene aggiornato con tutti i metadati e i riferimenti incrociati dei documenti letti, come etichette, nomi di intestazioni, oggetti Python descritti e voci di indice. Questo verrà successivamente utilizzato per sostituire i nodi temporanei.

I doctree analizzati vengono memorizzati sul disco, poiché non è possibile mantenerli tutti in memoria.

Fase 2: Verifiche di coerenza

Vengono eseguiti alcuni controlli per garantire che non vi siano sorprese nei documenti creati.

Fase 3: Risoluzione

Ora che i metadati e i dati dei riferimenti incrociati di tutti i documenti esistenti sono noti, tutti i nodi temporanei vengono sostituiti da nodi che possono essere convertiti in output utilizzando componenti chiamati trasformazioni. Ad esempio, vengono creati collegamenti per riferimenti a oggetti esistenti e vengono creati nodi letterali semplici per quelli che non esistono.

Fase 4: Scrittura

Questa fase converte i doctree risolti nel formato di output desiderato, come HTML o LaTeX. Ciò avviene tramite un cosiddetto docutils writer che visita i singoli nodi di ciascun doctree e produce alcuni output nel processo.

Nota: Alcuni builder si discostano da questo piano di costruzione generale, ad esempio, il builder che controlla i link [*collegamenti*] esterni non ha bisogno di altro che dei doctree analizzati e quindi non ha le fasi 2--4.

Per vedere un esempio di applicazione, fare riferimento a [Estendere il processo di build](#).

⁴⁰⁸ <https://docutils.sourceforge.io/docs/ref/doctree.html>

6.3 Metadati dell'estensione

Added in version 1.3.

La funzione `setup()` può restituire un dizionario. Questi vengono trattati da Sphinx come metadati dell'estensione. Le chiavi dei metadati attualmente riconosciute sono:

- `'version'`: una stringa che identifica la versione dell'estensione. Viene utilizzata per verificare i requisiti della versione dell'estensione (vedere [needs_extensions](#)) e per scopi informativi. Se non fornito, viene sostituito da `"unknown version"`.
- `'env_version'`: un numero intero che identifica la versione della struttura dati `env` se l'estensione memorizza dati nell'ambiente. Viene utilizzato per rilevare che la struttura dei dati è stata modificata rispetto all'ultima build. Le estensioni devono incrementare la versione quando la struttura dei dati è cambiata. Se non fornito, Sphinx considera che l'estensione non memorizza alcun dato nell'ambiente.
- `'parallel_read_safe'`: un valore booleano che specifica se è possibile utilizzare la lettura parallela dei file sorgenti quando viene caricata l'estensione. Il default è `False`, ovvero si deve specificare esplicitamente che l'estensione è sicura in lettura parallela dopo aver verificato che lo sia.

Nota: L'estensione *parallel-read-safe* deve soddisfare le seguenti condizioni:

- La logica centrale dell'estensione è eseguibile parallelamente durante la fase di lettura.
- Dispone di gestori di eventi per gli eventi *env-merge-info* e *env-purge-doc* se memorizza i dati nell'oggetto dell'ambiente di build (`env`) durante la fase di lettura.

-
- `'parallel_write_safe'`: un valore booleano che specifica se è possibile utilizzare la scrittura parallela dei file di output quando viene caricata l'estensione. Poiché le estensioni di solito non influenzano negativamente il processo, il valore di default è `True`.

Nota: L'estensione *parallel-write-safe* deve soddisfare le seguenti condizioni:

- La logica centrale dell'estensione è eseguibile parallelamente durante la fase di scrittura.
-

6.4 API utilizzate per scrivere le estensioni

Queste sezioni forniscono una descrizione più completa degli strumenti a disposizione durante lo sviluppo delle estensioni Sphinx. Alcuni sono fondamentali per Sphinx (come [API dell'applicazione](#)) mentre altri attivano comportamenti specifici (come [API i18n](#))

API dell'applicazione

Ogni estensione Sphinx è un modulo Python con almeno una funzione `setup()` function. Tale funzione viene chiamata al momento dell'inizializzazione con un argomento, l'oggetto dell'applicazione che rappresenta il processo Sphinx.

`class sphinx.application.Sphinx`

Questo oggetto applicazione dispone dell'API pubblica descritta di seguito.

Configurazione dell'estensione

Questi metodi vengono solitamente richiamati nella funzione `setup()`.

Esempi di utilizzo dell'API dell'estensione Sphinx possono essere visti nel pacchetto `sphinx.ext`.

`Sphinx.setup_extension(extname: str409) → None410`

Importa e configura un modulo di estensione Sphinx.

Carica l'estensione data dal modulo *name*. Lo si usa se l'estensione necessita delle funzionalità fornite da un'altra estensione. No-op se chiamato due volte.

`static Sphinx.require_sphinx(version: tuple411[int412, int413] | str414) → None415`

Controlla la versione di Sfinxe se richiesto.

Confronta *version* con la versione in esecuzione di Sphinx, e interrompe la build quando è troppo vecchia.

Parametri

version -- La versione richiesta nel formato `major.minor` o `(major, minor)`.

Added in version 1.0.

Cambiato nella versione 7.1: Il tipo di *versione* ora consente la forma `(major, minor)`.

`Sphinx.connect(event: str416, callback: Callable417, priority: int418 = 500) → int419`

Registra la *callback* per essere chiamata quando viene emesso *event*.

Per dettagli sugli eventi "core" disponibili e sugli argomenti delle funzioni di callback, vedere [API di callback degli eventi](#).

Parametri

- **event** -- Il nome dell'evento target
- **callback** -- Funzione di callback per l'evento

⁴⁰⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁴¹⁰ <https://docs.python.org/3/library/constants.html#None>

⁴¹¹ <https://docs.python.org/3/library/stdtypes.html#tuple>

⁴¹² <https://docs.python.org/3/library/functions.html#int>

⁴¹³ <https://docs.python.org/3/library/functions.html#int>

⁴¹⁴ <https://docs.python.org/3/library/stdtypes.html#str>

⁴¹⁵ <https://docs.python.org/3/library/constants.html#None>

- **priority** -- La priorità della callback. Le callback verranno richiamate in ordine di *priorità* (ascendente).

Ritorna

Un ID di listener. Può essere utilizzato per `disconnect()`.

Cambiato nella versione 3.0: Supporto di *priority*

`Sphinx.disconnect(listener_id: int420) → None421`

De-registra la callback dal `listener_id`.

Parametri

listener_id -- Un listener_id restituito da `connect()`

`Sphinx.add_builder(builder: type422[Builder], override: bool423 = False) → None424`

Registra un nuovo builder.

Parametri

- **builder** -- Una classe builder
- **override** -- Se true, installa forzatamente il builder anche se è già installato un altro builder con lo stesso nome

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

`Sphinx.add_config_value(name: str425, default: Any426, rebuild: Literal427['', 'env', 'epub', 'gettext', 'html', 'applehelp', 'devhelp'], types: type428 | Collection429[type430] | ENUM = ()) → None431`

Registra un valore di configurazione.

Ciò è necessario affinché Sphinx riconosca i nuovi valori e imposta di conseguenza i valori di default.

Parametri

- **name** -- Il nome del valore della configurazione. Si consiglia di anteporre il nome dell'estensione (es. `html_logo`, `epub_title`)
- **default** -- Il valore di default della configurazione.
- **rebuild** -- La condizione della re-build. Deve essere uno di questi valori: * `'env'` se una modifica nell'impostazione ha effetto solo quando un documento viene analizzato -- ciò significa che l'intero ambiente deve essere ricostruito. * `'html'` se una modifica nell'impostazione richiede una re-build completa dei documenti

⁴¹⁶ <https://docs.python.org/3/library/stdtypes.html#str>

⁴¹⁷ <https://docs.python.org/3/library/typing.html#typing.Callable>

⁴¹⁸ <https://docs.python.org/3/library/functions.html#int>

⁴¹⁹ <https://docs.python.org/3/library/functions.html#int>

⁴²⁰ <https://docs.python.org/3/library/functions.html#int>

⁴²¹ <https://docs.python.org/3/library/constants.html#None>

⁴²² <https://docs.python.org/3/library/functions.html#type>

⁴²³ <https://docs.python.org/3/library/functions.html#bool>

⁴²⁴ <https://docs.python.org/3/library/constants.html#None>

HTML. * ' ' se una modifica nell'impostazione non richiederà alcuna re-build speciale.

- **types** -- Il tipo di valore di configurazione. È possibile specificare un elenco di tipi. Per esempio, [str] viene utilizzato per descrivere una configurazione che accetta un valore stringa.

Cambiato nella versione 0.4: Se il valore di *default* è callable, verrà chiamato con l'oggetto config come argomento per ottenere il valore di default. Questo è utilizzabile per implementare valori di configurazione il cui valore di default dipende da altri valori.

Cambiato nella versione 0.6: Modificato *rebuild* da un semplice booleano (equivalente a ' ' o a 'env') a una stringa. Tuttavia, i valori booleani vengono ancora accettati e convertiti internamente.

Sphinx.add_event(*name*: str⁴³²) → None⁴³³

Registra un evento chiamato *name*.

Questo è necessario per poterlo emettere.

Parametri

name -- Il nome dell'evento

Sphinx.set_translator(*name*: str⁴³⁴, *translator_class*: type⁴³⁵[nodes.NodeVisitor], *override*: bool⁴³⁶ = False) → None⁴³⁷

Registra o sovrascrive una classe traduttore Docutils.

Viene utilizzata per registrare un traduttore di output personalizzato o per sostituirne uno nativo. Questo consente alle estensioni di utilizzare un traduttore personalizzato e definire nodi personalizzati per il traduttore (vedere [add_node\(\)](#)).

Parametri

- **name** -- Il nome del builder per il traduttore
- **translator_class** -- Una classe traduttore
- **override** -- Se true, installa forzatamente il traduttore anche se è già installato un altro traduttore con lo stesso nome

Added in version 1.3.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

⁴²⁵ <https://docs.python.org/3/library/stdtypes.html#str>

⁴²⁶ <https://docs.python.org/3/library/typing.html#typing.Any>

⁴²⁷ <https://docs.python.org/3/library/typing.html#typing.Literal>

⁴²⁸ <https://docs.python.org/3/library/functions.html#type>

⁴²⁹ <https://docs.python.org/3/library/collections.abc.html#collections.abc.Collection>

⁴³⁰ <https://docs.python.org/3/library/functions.html#type>

⁴³¹ <https://docs.python.org/3/library/constants.html#None>

⁴³² <https://docs.python.org/3/library/stdtypes.html#str>

⁴³³ <https://docs.python.org/3/library/constants.html#None>

⁴³⁴ <https://docs.python.org/3/library/stdtypes.html#str>

⁴³⁵ <https://docs.python.org/3/library/functions.html#type>

⁴³⁶ <https://docs.python.org/3/library/functions.html#bool>

⁴³⁷ <https://docs.python.org/3/library/constants.html#None>

`Sphinx.add_node(node: type438[Element], override: bool439 = False, **kwargs: tuple440[Callable, Callable | None441]) → None442`

Registra una classe di nodo Docutils.

Ciò è necessario per le attività interne di Docutils. Potrebbe anche essere utilizzato in futuro per convalidare i nodi nei documenti analizzati.

Parametri

- **node** -- Una classe nodo
- **kwargs** -- Funzioni "visitor" per ogni costruttore (vedere sotto)
- **override** -- Se true, installa il nodo forzatamente anche se è già installato un altro nodo con lo stesso nome

Le funzioni visitor del nodo per i writer Sphinx HTML, LaTeX, testo e pagine man possono essere fornite come argomenti di parole chiave: la parola chiave dovrebbe essere uno o più tra 'html', 'latex', 'text', 'man', 'texinfo' o qualsiasi altro traduttore supportato, il valore è una tupla di 2 metodi (visit, depart). depart può essere None se la funzione visit solleva docutils.nodes.SkipNode. Esempio:

```
class math(docutils.nodes.Element): pass

def visit_math_html(self, node):
    self.body.append(self.starttag(node, 'math'))
def depart_math_html(self, node):
    self.body.append('</math>')

app.add_node(math, html=(visit_math_html, depart_math_html))
```

Ovviamente, i traduttori per i quali non si specificano i metodi visitor si strozzeranno sul nodo quando incontrati in un documento da tradurre.

Cambiato nella versione 0.5: Aggiunto il supporto per argomenti di parole chiave che forniscono funzioni "visit".

`Sphinx.add_enumerable_node(node: type443[Element], figtype: str444, title_getter: TitleGetter | None445 = None, override: bool446 = False, **kwargs: tuple447[Callable, Callable]) → None448`

Registra una classe di nodo Docutils come target di numfig.

Sphinx numera il nodo automaticamente. E poi gli utenti possono fare riferimento ad esso utilizzando [numref](#).

Parametri

- **node** -- Una classe nodo

⁴³⁸ <https://docs.python.org/3/library/functions.html#type>

⁴³⁹ <https://docs.python.org/3/library/functions.html#bool>

⁴⁴⁰ <https://docs.python.org/3/library/stdtypes.html#tuple>

⁴⁴¹ <https://docs.python.org/3/library/constants.html#None>

⁴⁴² <https://docs.python.org/3/library/constants.html#None>

- **figtype** -- Il tipo dei nodi enumerabili. Ogni figtype ha sequenze di numerazione individuali. Come tipi di rappresentazione del sistema vengono definiti `figure`, `table` e `code-block`. È possibile aggiungere nodi personalizzati a questi figtype di default. È anche possibile definire un nuovo figtype personalizzato se viene fornito un nuovo figtype.
- **title_getter** -- Una funzione getter per ottenere il titolo del nodo. Richiede un'istanza del nodo enumerabile e deve restituire il titolo come stringa. Il titolo viene utilizzato come titolo di default dei riferimenti per [ref](#). Per default, Sphinx cerca `docutils.nodes.caption` o `docutils.nodes.title` nel nodo come titolo.
- **kwargs** -- Funzioni visitor per ciascun builder (lo stesso di `add_node()`)
- **override** -- Se true, installa il nodo forzatamente anche se è già installato un altro nodo con lo stesso nome

Added in version 1.4.

`Sphinx.add_directive(name: str449, cls: type450[Directive], override: bool451 = False)`
 → [None](#)⁴⁵²

Registra una direttiva Docutils.

Parametri

- **name** -- Il nome della direttiva
- **cls** -- Una classe direttiva
- **override** -- Se false, non la installa se è già installata un'altra direttiva con lo stesso nome. Se vero, installa incondizionatamente la direttiva.

Per esempio, una direttiva custom chiamata `my-directive` verrebbe aggiunta in questo modo:

```
from docutils.parsers.rst import Directive, directives

class MyDirective(Directive):
    has_content = True
    required_arguments = 1
    optional_arguments = 0
    final_argument_whitespace = True
    option_spec = {
        'class': directives.class_option,
```

(continues on next page)

⁴⁴³ <https://docs.python.org/3/library/functions.html#type>

⁴⁴⁴ <https://docs.python.org/3/library/stdtypes.html#str>

⁴⁴⁵ <https://docs.python.org/3/library/constants.html#None>

⁴⁴⁶ <https://docs.python.org/3/library/functions.html#bool>

⁴⁴⁷ <https://docs.python.org/3/library/stdtypes.html#tuple>

⁴⁴⁸ <https://docs.python.org/3/library/constants.html#None>

(continua dalla pagina precedente)

```

        'name': directives.unchanged,
    }

    def run(self):
        ...

def setup(app):
    app.add_directive('my-directive', MyDirective)

```

Per maggiori dettagli, vedere [i doc Docutils](#)⁴⁵³.

Cambiato nella versione 0.6: Le classi direttive in stile Docutils 0.5 ora sono supportate.

Deprecato dalla versione 1.8: Il supporto delle direttive in stile Docutils 0.4 (basate su funzioni) è deprecato.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

Sphinx.add_role(name: *str*⁴⁵⁴, role: *Any*⁴⁵⁵, override: *bool*⁴⁵⁶ = *False*) → *None*⁴⁵⁷

Registra un ruolo Docutils.

Parametri

- **name** -- Il nome del ruolo
- **role** -- Una funzione ruolo
- **override** -- Se *false*, non lo installa se è già installato un altro ruolo con lo stesso nome. Se *vero*, installa il ruolo incondizionatamente.

Per maggiori dettagli sulle funzioni dei ruoli, vedere [i doc Docutils](#)⁴⁵⁸.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

Sphinx.add_generic_role(name: *str*⁴⁵⁹, nodeclass: *Any*⁴⁶⁰, override: *bool*⁴⁶¹ = *False*) → *None*⁴⁶²

Registra un ruolo Docutils generico.

Registra un ruolo Docutils che non fa altro che avvolgere il suo contenuto nel nodo dato da *nodeclass*.

Parametri

- **override** -- Se *false*, non lo installa se è già installato un altro ruolo con lo stesso nome. Se *vero*, installa il ruolo incondizionatamente.

⁴⁴⁹ <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁵⁰ <https://docs.python.org/3/library/functions.html#type>
⁴⁵¹ <https://docs.python.org/3/library/functions.html#bool>
⁴⁵² <https://docs.python.org/3/library/constants.html#None>
⁴⁵³ <https://docutils.sourceforge.io/docs/howto/rst-directives.html>
⁴⁵⁴ <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁵⁵ <https://docs.python.org/3/library/typing.html#typing.Any>
⁴⁵⁶ <https://docs.python.org/3/library/functions.html#bool>
⁴⁵⁷ <https://docs.python.org/3/library/constants.html#None>
⁴⁵⁸ <https://docutils.sourceforge.io/docs/howto/rst-roles.html>

Added in version 0.6.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

`Sphinx.add_domain(domain: type463[Domain], override: bool464 = False) → None465`

Registra un dominio.

Parametri

- **domain** -- Una classe dominio
- **override** -- Se false, non lo installa se è già installato un altro dominio con lo stesso nome. Se vero, installa il dominio incondizionatamente.

Added in version 1.0.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

`Sphinx.add_directive_to_domain(domain: str466, name: str467, cls: type468[Directive], override: bool469 = False) → None470`

Registra una direttiva Docutils in un dominio.

Come [add_directive\(\)](#), ma la direttiva viene aggiunta al dominio denominato *domain*.

Parametri

- **domain** -- Il nome del dominio target
- **name** -- Un nome di direttiva
- **cls** -- Una classe direttiva
- **override** -- Se false, non la installa se è già installata un'altra direttiva con lo stesso nome. Se vero, installa incondizionatamente la direttiva.

Added in version 1.0.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

`Sphinx.add_role_to_domain(domain: str471, name: str472, role: RoleFunction | XRefRole, override: bool473 = False) → None474`

Registra un ruolo Docutils in un dominio.

Come [add_role\(\)](#), ma il ruolo viene aggiunto al dominio denominato *domain*.

⁴⁵⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁴⁶⁰ <https://docs.python.org/3/library/typing.html#typing.Any>

⁴⁶¹ <https://docs.python.org/3/library/functions.html#bool>

⁴⁶² <https://docs.python.org/3/library/constants.html#None>

⁴⁶³ <https://docs.python.org/3/library/functions.html#type>

⁴⁶⁴ <https://docs.python.org/3/library/functions.html#bool>

⁴⁶⁵ <https://docs.python.org/3/library/constants.html#None>

⁴⁶⁶ <https://docs.python.org/3/library/stdtypes.html#str>

⁴⁶⁷ <https://docs.python.org/3/library/stdtypes.html#str>

⁴⁶⁸ <https://docs.python.org/3/library/functions.html#type>

⁴⁶⁹ <https://docs.python.org/3/library/functions.html#bool>

⁴⁷⁰ <https://docs.python.org/3/library/constants.html#None>

Parametri

- **domain** -- Il nome del dominio target
- **name** -- Il nome del ruolo
- **role** -- La funzione ruolo
- **override** -- Se false, non lo installa se è già installato un altro ruolo con lo stesso nome. Se vero, installa il ruolo incondizionatamente.

Added in version 1.0.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

`Sphinx.add_index_to_domain(domain: str475, index: type476[Index], override: bool477 = False) → None478`

Registra un indice personalizzato per un dominio.

Aggiungi una classe *index* personalizzata al dominio denominata *domain*.

Parametri

- **domain** -- Il nome del dominio target
- **index** -- La classe index
- **override** -- Se false, non lo installa se è già installato un altro indice con lo stesso nome. Se vero, installa l'indice incondizionatamente.

Added in version 1.0.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

`Sphinx.add_object_type(directivename: str479, rolename: str480, indextemplate: str481 = "", parse_node: Callable482 | None483 = None, ref_nodeclass: type484[TextElement] | None485 = None, objname: str486 = "", doc_field_types: Sequence487 = (), override: bool488 = False) → None489`

Registra un nuovo tipo di oggetto.

Questo metodo è un modo molto conveniente per aggiungere un nuovo tipo *object* a cui è possibile fare riferimenti incrociati. Farà questo:

- Crea una nuova direttiva (chiamata *directivename*) per documentare un oggetto. Aggiungerà automaticamente le voci dell'indice se *indextemplate* non è vuoto; se fornito, deve contenere esattamente un'istanza di %s. Vedere l'esempio seguente per come verrà interpretato il template.

⁴⁷¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁴⁷² <https://docs.python.org/3/library/stdtypes.html#str>

⁴⁷³ <https://docs.python.org/3/library/functions.html#bool>

⁴⁷⁴ <https://docs.python.org/3/library/constants.html#None>

⁴⁷⁵ <https://docs.python.org/3/library/stdtypes.html#str>

⁴⁷⁶ <https://docs.python.org/3/library/functions.html#type>

⁴⁷⁷ <https://docs.python.org/3/library/functions.html#bool>

⁴⁷⁸ <https://docs.python.org/3/library/constants.html#None>

- Crea un nuovo ruolo (chiamato *rolename*) per fare riferimenti incrociati a queste descrizioni di oggetti.
- Fornendo *parse_node*, deve essere una funzione che accetta una stringa e un nodo *docutils* e deve popolare il nodo con i figli analizzati dalla stringa. Deve poi restituire il nome dell'elemento da utilizzare nei riferimenti incrociati e nelle voci di indice. Vedere il file `conf.py` nel sorgente di questa documentazione per un esempio.
- *objname* (se non fornito, verrà impostato automaticamente su *directivename*) nomina il tipo di oggetto. Viene utilizzato quando si elencano oggetti, ad es. nei risultati della ricerca.

Per esempio, se si ha questa chiamata in un'estensione Sphinx personalizzata:

```
app.add_object_type('directive', 'dir', 'pair: %s; directive')
```

si può utilizzare questo markup nei documenti:

```
.. rst:directive:: function

    Document a function.

<...>

See also the :rst:dir:`function` directive.
```

Per la direttiva, verrà generata una voce di indice come se fosse anteposto:

```
.. index:: pair: function; directive
```

Il nodo di riferimento sarà della classe `literal` (quindi verrà visualizzato in un carattere proporzionale, come appropriato per il codice) a meno che non venga fornito l'argomento *ref_nodeclass*, che deve essere una classe del nodo *docutils*. I più utili sono `docutils.nodes.emphasis` e `docutils.nodes.strong` -- si può usare anche `docutils.nodes.generated` se non si vogliono ulteriori decorazioni del testo. Se il testo deve essere trattato come letterale (ad esempio senza sostituzione delle virgolette), ma non ha lo stile della "macchina da scrivere", utilizzare `sphinx.addnodes.literal_emphasis` o `sphinx.addnodes.literal_strong`.

Per il contenuto del ruolo, si hanno le stesse possibilità sintattiche dei ruoli Sphinx standard (vedere *Sintassi dei riferimenti incrociati*).

Se *override* è `True`, l'*object_type* specificato viene installato forzatamente anche se è già installato un *object_type*.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

`Sphinx.add_crossref_type`(*directivename*: *str*⁴⁹⁰, *rolename*: *str*⁴⁹¹, *indextemplate*: *str*⁴⁹²
 = "", *ref_nodeclass*: *type*⁴⁹³[*TextElement*] | *None*⁴⁹⁴ = *None*,
objname: *str*⁴⁹⁵ = "", *override*: *bool*⁴⁹⁶ = *False*) → *None*⁴⁹⁷

Registra un nuovo tipo di oggetto crossref.

Questo metodo è molto simile a `add_object_type()` tranne per il fatto che la direttiva che genera deve essere vuota e non produrrà alcun output.

Ciò significa che si possono aggiungere target semantici ai sorgenti e fare riferimento ad essi utilizzando ruoli personalizzati anziché generici (come `ref`). Esempio di chiamata:

```
app.add_crossref_type('topic', 'topic', 'single: %s',
                     docutils.nodes.emphasis)
```

Esempio d'uso:

```
.. topic:: application API

The application API
-----

Some random text here.

See also :topic:`this section <application API>`.
```

(Naturalmente, l'elemento che segue la direttiva `topic` non deve essere necessariamente una sezione).

Parametri

override -- Se *false*, non lo installa se è già installato un altro tipo di riferimento incrociato con lo stesso nome. Se *true*, installa incondizionatamente il tipo di riferimento incrociato.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

⁴⁷⁹ <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁸⁰ <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁸¹ <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁸² <https://docs.python.org/3/library/typing.html#typing.Callable>
⁴⁸³ <https://docs.python.org/3/library/constants.html#None>
⁴⁸⁴ <https://docs.python.org/3/library/functions.html#type>
⁴⁸⁵ <https://docs.python.org/3/library/constants.html#None>
⁴⁸⁶ <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁸⁷ <https://docs.python.org/3/library/collections.abc.html#collections.abc.Sequence>
⁴⁸⁸ <https://docs.python.org/3/library/functions.html#bool>
⁴⁸⁹ <https://docs.python.org/3/library/constants.html#None>
⁴⁹⁰ <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁹¹ <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁹² <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁹³ <https://docs.python.org/3/library/functions.html#type>
⁴⁹⁴ <https://docs.python.org/3/library/constants.html#None>
⁴⁹⁵ <https://docs.python.org/3/library/stdtypes.html#str>
⁴⁹⁶ <https://docs.python.org/3/library/functions.html#bool>
⁴⁹⁷ <https://docs.python.org/3/library/constants.html#None>

`Sphinx.add_transform(transform: type498[Transform]) → None499`

Registra una trasformazione Docutils da applicare dopo l'analisi.

Aggiunge la sottoclasse standard docutils Transform *transform* alla lista delle trasformazioni applicate dopo che Sphinx ha analizzato un documento reST.

Parametri

transform -- Una classe transform

Tabella 1: gamma di priorità per le trasformazioni Sphinx

Priorità	Uso principale in Sphinx
0-99	Corregge i nodi non validi tramite docutils. Traduce un doctree.
100-299	Preparazione
300-399	prima
400-699	main
700-799	Post elaborazione. Scadenza per modificare testo e riferimenti.
800-899	Raccogli riferimenti e nodi referenziati. Elaborazione del dominio.
900-999	Finalizza e ripulisce.

refs: [Transform Priority Range Categories](#)⁵⁰⁰

`Sphinx.add_post_transform(transform: type501[Transform]) → None502`

Registra una trasformazione Docutils da applicare prima della scrittura.

Aggiunge la sottoclasse standard dei documenti Transform *transform* alla lista delle trasformazioni che vengono applicate prima che Sphinx scriva un documento.

Parametri

transform -- Una classe transform

`Sphinx.add_js_file(filename: str503 | None504, priority: int505 = 500, loading_method: str506 | None507 = None, **kwargs: Any508) → None509`

Registra un file JavaScript da includere nell'output HTML.

Parametri

- **filename** -- Il nome di un file JavaScript che includerà il template HTML di default. Deve essere relativo al path statico HTML, o un URI completo con schema, o `None`. Il valore `None` viene utilizzato per creare un tag in linea `<script>`. Vedere la descrizione di *kwargs* di seguito.
- **priority** -- I file sono inclusi in ordine crescente di priorità. Se più file JavaScript hanno la stessa priorità, tali file verranno inclusi in ordine di registrazione. Consultare la lista degli "intervalli di priorità per i file JavaScript" di seguito.

⁴⁹⁸ <https://docs.python.org/3/library/functions.html#type>

⁴⁹⁹ <https://docs.python.org/3/library/constants.html#None>

⁵⁰⁰ <https://docutils.sourceforge.io/docs/ref/transforms.html#transform-priority-range-categories>

⁵⁰¹ <https://docs.python.org/3/library/functions.html#type>

⁵⁰² <https://docs.python.org/3/library/constants.html#None>

- **loading_method** -- Il metodo di caricamento per il file JavaScript. Sono consentiti sia 'async' che 'defer'.
- **kwargs** -- Argomenti aggiuntivi per le parole chiave sono inclusi come attributi del tag <script>. Se viene fornita la parola chiave speciale body, il suo valore verrà aggiunto come contenuto del tag <script>.

Esempio:

```
app.add_js_file('example.js')
# => <script src="_static/example.js"></script>

app.add_js_file('example.js', loading_method="async")
# => <script src="_static/example.js" async="async"></script>

app.add_js_file(None, body="var myVariable = 'foo';")
# => <script>var myVariable = 'foo';</script>
```

Tabella 2: intervalli di priorità per i file JavaScript

Priorità	Uso principale in Sphinx
200	priorità di default per i file JavaScript nativi
500	priorità di default per le estensioni
800	priorità di default per html_js_files

Un file JavaScript può essere aggiunto alla pagina HTML specifica quando un'estensione chiama questo metodo sull'evento [html-page-context](#).

Added in version 0.5.

Cambiato nella versione 1.8: Rinominato da `app.add_javascript()`. E consente argomenti di parole chiave come gli attributi del tag script.

Cambiato nella versione 3.5: Prende l'argomento prioritario. Consente di aggiungere un file JavaScript alla pagina specifica.

Cambiato nella versione 4.4: Prende l'argomento `loading_method`. Consente di modificare il metodo di caricamento del file JavaScript.

`Sphinx.add_css_file(filename: str510, priority: int511 = 500, **kwargs: Any512) → None513`

Registra un foglio di stile da includere nell'output HTML.

Parametri

⁵⁰³ <https://docs.python.org/3/library/stdtypes.html#str>
⁵⁰⁴ <https://docs.python.org/3/library/constants.html#None>
⁵⁰⁵ <https://docs.python.org/3/library/functions.html#int>
⁵⁰⁶ <https://docs.python.org/3/library/stdtypes.html#str>
⁵⁰⁷ <https://docs.python.org/3/library/constants.html#None>
⁵⁰⁸ <https://docs.python.org/3/library/typing.html#typing.Any>
⁵⁰⁹ <https://docs.python.org/3/library/constants.html#None>

- **filename** -- Il nome di un file CSS che includerà il template HTML di default. Deve essere relativo al path statico HTML o a un URI completo con schema.
- **priority** -- I file sono inclusi in ordine crescente di priorità. Se più file CSS hanno la stessa priorità, tali file verranno inclusi in ordine di registrazione. Consultare la lista degli "intervalli di priorità per i file CSS" di seguito.
- **kwargs** -- Argomenti aggiuntivi per le parole chiave vengono inclusi come attributi del tag `<link>`.

Esempio:

```
app.add_css_file('custom.css')
# => <link rel="stylesheet" href="_static/custom.css" type="text/
#      ↪css" />

app.add_css_file('print.css', media='print')
# => <link rel="stylesheet" href="_static/print.css"
#      type="text/css" media="print" />

app.add_css_file('fancy.css', rel='alternate stylesheet', title=
#      ↪'fancy')
# => <link rel="alternate stylesheet" href="_static/fancy.css"
#      type="text/css" title="fancy" />
```

Tabella 3: intervallo di priorità per i file CSS

Priorità	Uso principale in Sphinx
200	priorità di default per i file CSS nativi
500	priorità di default per le estensioni
800	priorità di default per html_css_files

Un file CSS può essere aggiunto alla pagina HTML specifica quando un'estensione chiama questo metodo sull'evento [html-page-context](#).

Added in version 1.0.

Cambiato nella versione 1.6: Gli attributi opzionali *alternate* e/o *title* possono essere forniti con gli argomenti *alternate* (un Booleano) e *title* (una stringa). Il default è senza titolo e *alternate* = False. Per ulteriori informazioni, fare riferimento alla [documentazione](#)⁵¹⁴.

Cambiato nella versione 1.8: Rinominato da `app.add_stylesheet()`. E consente argomenti di parole chiave come attributi del tag dei link.

Cambiato nella versione 3.5: Prende l'argomento prioritario. Consente di aggiungere un file CSS alla pagina specifica.

Sphinx.add_latex_package(*packagename*: [str](#)⁵¹⁵, *options*: [str](#)⁵¹⁶ | [None](#)⁵¹⁷ = [None](#),
after_hyperref: [bool](#)⁵¹⁸ = [False](#)) → [None](#)⁵¹⁹

Registra un pacchetto da includere nel codice sorgente LaTeX.

Aggiunge *packagename* alla lista dei pacchetti che il codice sorgente LaTeX includerà. Fornendo *options*, queste verranno indirizzate alla dichiarazione *usepackage*. Impostando *after_hyperref* a true, il pacchetto verrà caricato dopo il pacchetto *hyperref*.

```
app.add_latex_package('mypackage')
# => \usepackage{mypackage}
app.add_latex_package('mypackage', 'foo,bar')
# => \usepackage[foo,bar]{mypackage}
```

Added in version 1.3.

Added in version 3.1: l'opzione *after_hyperref*.

Sphinx.add_lexer(*alias*: [str](#)⁵²⁰, *lexer*: [type](#)⁵²¹[[Lexer](#)]) → [None](#)⁵²²

Registra un nuovo lexer per il codice sorgente.

Usa *lexer* per evidenziare i blocchi di codice con l'*alias* del linguaggio specificato.

Added in version 0.6.

Cambiato nella versione 2.1: Prende una classe lexer come argomento.

Cambiato nella versione 4.0: Rimosso il supporto per le istanze lexer come argomento.

Sphinx.add_autodocumenter(*cls*: [Any](#)⁵²³, *override*: [bool](#)⁵²⁴ = [False](#)) → [None](#)⁵²⁵

Registra una nuova classe documenter per l'estensione autodoc.

Aggiunge *cls* come nuova classe documenter per l'estensione [sphinx.ext.autodoc](#). Deve essere una sottoclasse di [sphinx.ext.autodoc.Documenter](#). Questo consente di documentare automaticamente nuovi tipi di oggetti. Vedere il sorgente del modulo autodoc per esempi su come creare una sottoclasse di [Documenter](#).

Se *override* è True, il *cls* specificato viene installato forzatamente anche se è già installato un documenter con lo stesso nome.

Vedere [Sviluppo di estensioni autodoc](#).

Added in version 0.6.

⁵¹⁰ <https://docs.python.org/3/library/stdtypes.html#str>
⁵¹¹ <https://docs.python.org/3/library/functions.html#int>
⁵¹² <https://docs.python.org/3/library/typing.html#typing.Any>
⁵¹³ <https://docs.python.org/3/library/constants.html#None>
⁵¹⁴ https://mdn.io/Web/CSS/Alternative_style_sheets
⁵¹⁵ <https://docs.python.org/3/library/stdtypes.html#str>
⁵¹⁶ <https://docs.python.org/3/library/stdtypes.html#str>
⁵¹⁷ <https://docs.python.org/3/library/constants.html#None>
⁵¹⁸ <https://docs.python.org/3/library/functions.html#bool>
⁵¹⁹ <https://docs.python.org/3/library/constants.html#None>
⁵²⁰ <https://docs.python.org/3/library/stdtypes.html#str>
⁵²¹ <https://docs.python.org/3/library/functions.html#type>
⁵²² <https://docs.python.org/3/library/constants.html#None>

Cambiato nella versione 2.2: Aggiunta la parola chiave *override*.

`Sphinx.add_autodoc_attrgetter`(*typ*: [type](#)⁵²⁶, *getter*: [Callable](#)⁵²⁷[[[Any](#)⁵²⁸, [str](#)⁵²⁹, [Any](#)⁵³⁰], [Any](#)⁵³¹]) → [None](#)⁵³²

Registra una nuova funzione simile a `getattr` per l'estensione autodoc.

Aggiunge *getter*, che deve essere una funzione con un'interfaccia compatibile con il nativo `getattr()`⁵³³, come *getter* dell'attributo autodoc per oggetti che sono istanze di *typ*. Tutti i casi in cui autodoc necessita di ottenere un attributo di un tipo vengono quindi gestiti da questa funzione invece che da `getattr()`⁵³⁴.

Added in version 0.6.

`Sphinx.add_search_language`(*cls*: [Any](#)⁵³⁵) → [None](#)⁵³⁶

Registra un nuovo linguaggio per l'indice di ricerca HTML.

Aggiunge *cls*, che deve essere una sottoclasse di `sphinx.search.SearchLanguage`, come linguaggio di supporto per la creazione dell'indice di ricerca full-text HTML. La classe deve avere un attributo *lang* che indica il linguaggio per cui dovrebbe essere utilizzata. Vedere [html_search_language](#).

Added in version 1.1.

`Sphinx.add_source_suffix`(*suffix*: [str](#)⁵³⁷, *filetype*: [str](#)⁵³⁸, *override*: [bool](#)⁵³⁹ = *False*) → [None](#)⁵⁴⁰

Registra un suffisso di file sorgente.

Uguale a [source_suffix](#). Gli utenti possono sovrascriverlo (*override*) utilizzando il settaggio della configurazione.

Parametri

override -- Se *false*, non lo installa, lo stesso suffisso è già installato. Se *true*, installa incondizionatamente il suffisso.

Added in version 1.8.

`Sphinx.add_source_parser`(*parser*: [type](#)⁵⁴¹[[Parser](#)], *override*: [bool](#)⁵⁴² = *False*) → [None](#)⁵⁴³

⁵²³ <https://docs.python.org/3/library/typing.html#typing.Any>

⁵²⁴ <https://docs.python.org/3/library/functions.html#bool>

⁵²⁵ <https://docs.python.org/3/library/constants.html#None>

⁵²⁶ <https://docs.python.org/3/library/functions.html#type>

⁵²⁷ <https://docs.python.org/3/library/typing.html#typing.Callable>

⁵²⁸ <https://docs.python.org/3/library/typing.html#typing.Any>

⁵²⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁵³⁰ <https://docs.python.org/3/library/typing.html#typing.Any>

⁵³¹ <https://docs.python.org/3/library/typing.html#typing.Any>

⁵³² <https://docs.python.org/3/library/constants.html#None>

⁵³³ <https://docs.python.org/3/library/functions.html#getattr>

⁵³⁴ <https://docs.python.org/3/library/functions.html#getattr>

⁵³⁵ <https://docs.python.org/3/library/typing.html#typing.Any>

⁵³⁶ <https://docs.python.org/3/library/constants.html#None>

⁵³⁷ <https://docs.python.org/3/library/stdtypes.html#str>

⁵³⁸ <https://docs.python.org/3/library/stdtypes.html#str>

⁵³⁹ <https://docs.python.org/3/library/functions.html#bool>

⁵⁴⁰ <https://docs.python.org/3/library/constants.html#None>

Registra una classe parser.

Parametri

override -- Se false, non lo installa se è già installato un altro parser per lo stesso suffisso. Se true, installa incondizionatamente il parser.

Added in version 1.4.

Cambiato nella versione 1.8: L'argomento *suffix* è deprecato. Accetta solo l'argomento *parser*. Usare invece l'API `add_source_suffix()` per registrare il suffisso.

Cambiato nella versione 1.8: Aggiunta la parola chiave *override*.

`Sphinx.add_env_collector(collector: type544[EnvironmentCollector]) → None545`

Registra una classe collector dell'ambiente (environment).

Fare riferimento a *Environment Collector API*.

Added in version 1.6.

`Sphinx.add_html_theme(name: str546, theme_path: str547) → None548`

Registra un tema HTML.

Il *name* è il nome del tema e *theme_path* è il path completo del tema (rif: *Distribuire il tema come pacchetto Python*).

Added in version 1.6.

`Sphinx.add_html_math_renderer(name: str549, inline_renderers: tuple550[Callable551, Callable552 | None553] | None554 = None, block_renderers: tuple555[Callable556, Callable557 | None558] | None559 = None) → None560`

Registra un renderer matematico per HTML.

Il *name* è un nome del renderer matematico. Sia *inline_renderers* che *block_renderers* sono usati come funzioni visitor per il writer HTML: il primo per il nodo matematico in linea (`nodes.math`), il secondo per il nodo matematico a blocchi (`nodes.math_block`). Per quanto riguarda le funzioni visitor, vedere `add_node()` per i dettagli.

Added in version 1.8.

⁵⁴¹ <https://docs.python.org/3/library/functions.html#type>

⁵⁴² <https://docs.python.org/3/library/functions.html#bool>

⁵⁴³ <https://docs.python.org/3/library/constants.html#None>

⁵⁴⁴ <https://docs.python.org/3/library/functions.html#type>

⁵⁴⁵ <https://docs.python.org/3/library/constants.html#None>

⁵⁴⁶ <https://docs.python.org/3/library/stdtypes.html#str>

⁵⁴⁷ <https://docs.python.org/3/library/stdtypes.html#str>

⁵⁴⁸ <https://docs.python.org/3/library/constants.html#None>

`Sphinx.add_message_catalog(catalog: str561, locale_dir: str562) → None563`

Registra un catalogo di messaggi.

Parametri

- **catalog** -- Il nome del catalogo
- **locale_dir** -- Il path di base del catalogo dei messaggi

Per maggiori dettagli, vedere `sphinx.locale.get_translation()`.

Added in version 1.8.

`Sphinx.is_parallel_allowed(typ: str564) → bool565`

Controlla se l'elaborazione parallela è consentita o meno.

Parametri

typ -- Un tipo di elaborazione; 'read' o 'write'.

exception sphinx.application.ExtensionError

Tutti questi metodi sollevano questa eccezione se qualcosa è andato storto con l'API di estensione.

Emissione di eventi

class sphinx.application.Sphinx

`emit(event: str566, *args: Any567, allowed_exceptions: tuple568[type569[Exception570], ...] = ()) → list571`

Emette *event* e passa gli *arguments* alle funzioni di callback.

Restituisce i valori restituiti di tutti i callback come lista. Non emette eventi Sphinx "core" nelle estensioni!

Parametri

- **event** -- Il nome dell'evento che verrà emesso

⁵⁴⁹ <https://docs.python.org/3/library/stdtypes.html#str>
⁵⁵⁰ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁵⁵¹ <https://docs.python.org/3/library/typing.html#typing.Callable>
⁵⁵² <https://docs.python.org/3/library/typing.html#typing.Callable>
⁵⁵³ <https://docs.python.org/3/library/constants.html#None>
⁵⁵⁴ <https://docs.python.org/3/library/constants.html#None>
⁵⁵⁵ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁵⁵⁶ <https://docs.python.org/3/library/typing.html#typing.Callable>
⁵⁵⁷ <https://docs.python.org/3/library/typing.html#typing.Callable>
⁵⁵⁸ <https://docs.python.org/3/library/constants.html#None>
⁵⁵⁹ <https://docs.python.org/3/library/constants.html#None>
⁵⁶⁰ <https://docs.python.org/3/library/constants.html#None>
⁵⁶¹ <https://docs.python.org/3/library/stdtypes.html#str>
⁵⁶² <https://docs.python.org/3/library/stdtypes.html#str>
⁵⁶³ <https://docs.python.org/3/library/constants.html#None>
⁵⁶⁴ <https://docs.python.org/3/library/stdtypes.html#str>
⁵⁶⁵ <https://docs.python.org/3/library/functions.html#bool>

- **args** -- Gli argomenti per l'evento
- **allowed_exceptions** -- L'elenco delle eccezioni consentite nelle callback

Cambiato nella versione 3.1: Aggiunto *allowed_exceptions* per specificare le eccezioni "path-through"

emit_firstresult(*event*: *str*⁵⁷², **args*: *Any*⁵⁷³, *allowed_exceptions*: *tuple*⁵⁷⁴[*type*⁵⁷⁵[*Exception*⁵⁷⁶], ...] = ()) → *Any*⁵⁷⁷

Emette *event* e passa gli *arguments* alle funzioni di callback.

Restituisce il risultato del primo callback che non restituisce None.

Parametri

- **event** -- Il nome dell'evento che verrà emesso
- **args** -- Gli argomenti per l'evento
- **allowed_exceptions** -- L'elenco delle eccezioni consentite nelle callback

Added in version 0.5.

Cambiato nella versione 3.1: Aggiunto *allowed_exceptions* per specificare le eccezioni "path-through"

Informazioni sul runtime di Sphinx

L'oggetto dell'applicazione fornisce inoltre informazioni di runtime come attributi.

Sphinx.project

Progetto target. Vedere *Project*.

Sphinx.sourcedir

Directory sorgente.

Sphinx.confdir

Directory contenente *conf.py*.

⁵⁶⁶ <https://docs.python.org/3/library/stdtypes.html#str>
⁵⁶⁷ <https://docs.python.org/3/library/typing.html#typing.Any>
⁵⁶⁸ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁵⁶⁹ <https://docs.python.org/3/library/functions.html#type>
⁵⁷⁰ <https://docs.python.org/3/library/exceptions.html#Exception>
⁵⁷¹ <https://docs.python.org/3/library/stdtypes.html#list>
⁵⁷² <https://docs.python.org/3/library/stdtypes.html#str>
⁵⁷³ <https://docs.python.org/3/library/typing.html#typing.Any>
⁵⁷⁴ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁵⁷⁵ <https://docs.python.org/3/library/functions.html#type>
⁵⁷⁶ <https://docs.python.org/3/library/exceptions.html#Exception>
⁵⁷⁷ <https://docs.python.org/3/library/typing.html#typing.Any>

Sphinx.doctreedir

Directory per la memorizzazione di doctree pickled.

Sphinx.outdir

Directory per l'archiviazione del documento creato.

Sphinx.fresh_env_used

True/False a seconda che sia stato creato un nuovo ambiente per questa build o None se l'ambiente non è stato ancora inizializzato.

Eventi core di Sphinx

Nota: Spostato in *API di callback degli eventi*.

Controllo della versione di Sphinx

Lo si usa per adattare l'estensione alle modifiche delle API in Sphinx.

sphinx.version_info = (7, 4, 0, 'beta', 0)

Informazioni sulla versione per un migliore uso a livello di programmazione.

Una tupla di cinque elementi; per la versione Sphinx 1.2.1 beta 3 questo sarebbe (1, 2, 1, 'beta', 3). Il quarto elemento può essere uno tra: alpha, beta, rc, final. final ha sempre 0 come ultimo elemento.

Added in version 1.2: Prima della versione 1.2, controlla la stringa `sphinx.__version__`.

L'oggetto Config

class sphinx.config.Config(*config: dict⁵⁷⁸[str⁵⁷⁹, Any⁵⁸⁰] | None⁵⁸¹ = None, overrides: dict⁵⁸²[str⁵⁸³, Any⁵⁸⁴] | None⁵⁸⁵ = None*)

Astrazione del file di configurazione.

L'oggetto Config rende disponibili i valori di tutte le opzioni di configurazione come attributi.

Viene esposto tramite gli attributi `Sphinx.config` e `sphinx.environment.BuildEnvironment.config`. Per esempio, per ottenere il valore di `language`, utilizzare `app.config.language` o `env.config.language`.

Il template bridge

`class sphinx.application.TemplateBridge`

Questa classe definisce l'interfaccia per un "template bridge", ovvero una classe che esegue il rendering dei template con un nome di template e un contesto.

init(*builder*: `Builder`, *theme*: `Theme` | `None`⁵⁸⁶ = `None`, *dirs*: `list`⁵⁸⁷[`str`⁵⁸⁸] | `None`⁵⁸⁹ = `None`) → `None`⁵⁹⁰

Chiamato dal builder per inizializzare il template di sistema.

builder è l'oggetto builder; probabilmente si vorrà dare un'occhiata al valore di `builder.config.templates_path`.

theme è un oggetto `sphinx.theming.Theme` o `None`; in quest'ultimo caso, *dirs* può essere una lista di directory fisse in cui cercare i template.

newest_template_mtime() → `float`⁵⁹¹

Chiamato dal builder per determinare se i file di output sono obsoleti a causa delle modifiche al template. Restituisce l'mtime del file template più recente che è stato modificato. L'implementazione di default restituisce `0`.

render(*template*: `str`⁵⁹², *context*: `dict`⁵⁹³) → `None`⁵⁹⁴

Chiamato dal builder per eseguire il rendering di un template fornito come filename con un contesto specificato (un dizionario Python).

render_string(*template*: `str`⁵⁹⁵, *context*: `dict`⁵⁹⁶) → `str`⁵⁹⁷

Chiamato dal builder per eseguire il rendering di un template fornito come una stringa con un contesto specificato (un dizionario Python).

⁵⁷⁸ <https://docs.python.org/3/library/stdtypes.html#dict>

⁵⁷⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁵⁸⁰ <https://docs.python.org/3/library/typing.html#typing.Any>

⁵⁸¹ <https://docs.python.org/3/library/constants.html#None>

⁵⁸² <https://docs.python.org/3/library/stdtypes.html#dict>

⁵⁸³ <https://docs.python.org/3/library/stdtypes.html#str>

⁵⁸⁴ <https://docs.python.org/3/library/typing.html#typing.Any>

⁵⁸⁵ <https://docs.python.org/3/library/constants.html#None>

⁵⁸⁶ <https://docs.python.org/3/library/constants.html#None>

⁵⁸⁷ <https://docs.python.org/3/library/stdtypes.html#list>

⁵⁸⁸ <https://docs.python.org/3/library/stdtypes.html#str>

⁵⁸⁹ <https://docs.python.org/3/library/constants.html#None>

⁵⁹⁰ <https://docs.python.org/3/library/constants.html#None>

⁵⁹¹ <https://docs.python.org/3/library/functions.html#float>

⁵⁹² <https://docs.python.org/3/library/stdtypes.html#str>

⁵⁹³ <https://docs.python.org/3/library/stdtypes.html#dict>

⁵⁹⁴ <https://docs.python.org/3/library/constants.html#None>

⁵⁹⁵ <https://docs.python.org/3/library/stdtypes.html#str>

⁵⁹⁶ <https://docs.python.org/3/library/stdtypes.html#dict>

⁵⁹⁷ <https://docs.python.org/3/library/stdtypes.html#str>

Eccezioni

exception sphinx.errors.SphinxError

Classe base per gli errori Sphinx.

Questa è la classe base per le eccezioni "interessanti". Quando viene sollevata una tale eccezione, Sphinx interromperà la build e presenterà la categoria e il messaggio dell'eccezione all'utente.

Si consiglia di derivare le estensioni da questa per i relativi errori personalizzati.

Le eccezioni *non* derivate da `SphinxError` vengono trattate come inaspettate e mostrate all'utente con una parte del traceback (e il traceback completo salvato in un file temporaneo).

category

Descrizione della "category", dell'eccezione, utilizzata per convertire l'eccezione in una stringa ("category: message"). Dovrebbe essere impostato di conseguenza nelle sottoclassi.

exception sphinx.errors.ConfigError

Errore della configurazione.

exception sphinx.errors.ExtensionError(*message: str*⁵⁹⁸, *orig_exc: Exception*⁵⁹⁹ | *None*⁶⁰⁰ = None, *modname: str*⁶⁰¹ | *None*⁶⁰² = None)

Errore dell'estensione.

exception sphinx.errors.ThemeError

Errore del tema.

exception sphinx.errors.VersionRequirementError

Errore di versione Sphinx incompatibile.

API di callback degli eventi

Collegare le funzioni di callback agli eventi è un modo semplice per estendere Sphinx, collegandosi al processo di build in vari punti.

Usare `Sphinx.connect()` nella funzione setup di un'estensione, o una funzione setup nei propri progetti `conf.py`, per connettere le funzioni agli eventi:

```
def source_read_handler(app, docname, source):
    print('do something here...')
```

(continues on next page)

⁵⁹⁸ <https://docs.python.org/3/library/stdtypes.html#str>

⁵⁹⁹ <https://docs.python.org/3/library/exceptions.html#Exception>

⁶⁰⁰ <https://docs.python.org/3/library/constants.html#None>

⁶⁰¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁶⁰² <https://docs.python.org/3/library/constants.html#None>

(continua dalla pagina precedente)

```
def setup(app):
    app.connect('source-read', source_read_handler)
```

Vedi anche:

Le estensioni possono aggiungere i propri eventi utilizzando `Sphinx.add_event()` e chiamandoli con `Sphinx.emit()` o con `Sphinx.emit_firstresult()`.

Panoramica degli eventi core

Di seguito è riportata una panoramica dell'evento principale [core] che si verifica durante una build.

```
1. event.config-initiated(app, config)
2. event.builder-initiated(app)
3. event.env-get-outdated(app, env, added, changed, removed)
4. event.env-before-read-docs(app, env, docnames)

for docname in docnames:
    5. event.env-purge-doc(app, env, docname)

    if doc changed and not removed:
        6. source-read(app, docname, source)
        7. run source parsers: text -> docutils.document
           - parsers can be added with the app.add_source_parser() API
        8. apply transforms based on priority: docutils.document ->
           ↳ docutils.document
           - event.doctree-read(app, doctree) is called in the middle of
           ↳ transforms,
           transforms come before/after this event depending on their
           ↳ priority.

9. event.env-merge-info(app, env, docnames, other)
   - if running in parallel mode, this event will be emitted for each
   ↳ process

10. event.env-updated(app, env)
11. event.env-get-updated(app, env)
12. event.env-check-consistency(app, env)

# The updated-docs list can be builder dependent, but generally
↳ includes all new/changed documents,
# plus any output from `env-get-updated`, and then all "parent"
↳ documents in the ToC tree
# For builders that output a single page, they are first joined into a
↳ single doctree before post-transforms
```

(continues on next page)

(continua dalla pagina precedente)

```
# or the doctree-resolved event is emitted
for docname in updated-docs:
    13. apply post-transforms (by priority): docutils.document -> docutils.document
    14. event.doctree-resolved(app, doctree, docname)
        - In the event that any reference nodes fail to resolve, the following may emit:
            - event.missing-reference(env, node, contnode)
            - event.warn-missing-reference(domain, node)

15. Generate output files
16. event.build-finished(app, exception)
```

Ecco anche un diagramma di flusso degli eventi, nel contesto del processo di build di Sphinx:

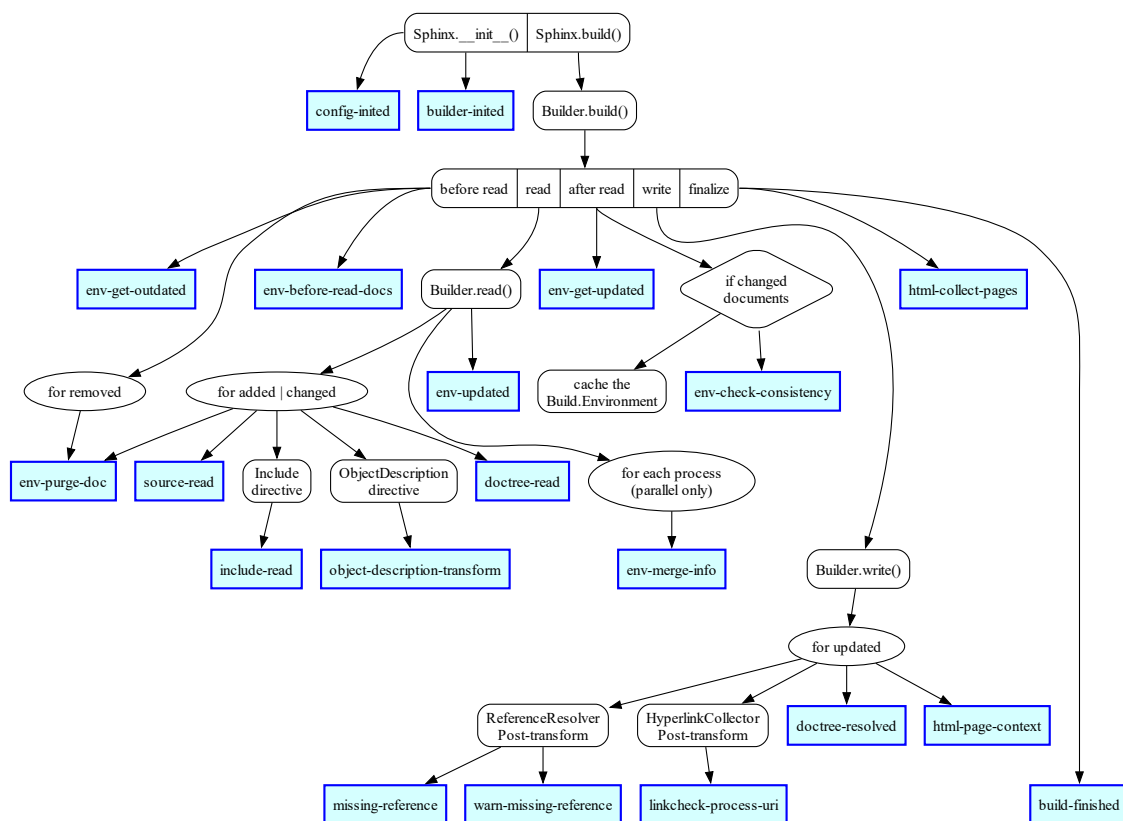


Fig. 1: Flusso degli eventi core di Sphinx

Dettagli sugli eventi core

Ecco un elenco più dettagliato di questi eventi.

config-inited(*app*, *config*)

Parametri

- **app** -- *Sphinx*
- **config** -- *Config*

Emesso quando l'oggetto config è stato inizializzato.

Added in version 1.8.

builder-inited(*app*)

Parametri

- **app** -- *Sphinx*

Emesso quando l'oggetto builder è stato creato (disponibile come `app.builder`).

env-get-outdated(*app*, *env*, *added*, *changed*, *removed*)

Parametri

- **app** -- *Sphinx*
- **env** -- *BuildEnvironment*
- **added** -- set[str]
- **changed** -- set[str]
- **removed** -- set[str]

restituisce

list[str] con i nomi dei documenti aggiuntivi da rileggere

Emesso quando l'ambiente determina quali file sorgenti sono cambiati e devono essere riletti. *added*, *changed* e *removed* sono insiemi di docname determinati dall'ambiente. Si può restituire un elenco di docname da rileggere oltre a questi.

Added in version 1.1.

env-purge-doc(*app*, *env*, *docname*)

Parametri

- **app** -- *Sphinx*
- **env** -- *BuildEnvironment*
- **docname** -- str

Emesso quando tutte le tracce [tracce] di un file sorgente devono essere rimosse dall'ambiente, ovvero se il file sorgente viene rimosso o prima che venga letto di nuovo. Questo è per le estensioni che mantengono le proprie cache negli attributi dell'ambiente.

Ad esempio, c'è una cache di tutti i moduli nell'ambiente. Quando un file sorgente viene modificato, le voci della cache per il file vengono cancellate, poiché le dichiarazioni del modulo potrebbero essere state rimosse dal file.

Added in version 0.5.

env-before-read-docs(*app*, *env*, *docnames*)

Parametri

- **app** -- *Sphinx*
- **env** -- *BuildEnvironment*
- **docnames** -- list[str]

Emesso dopo che l'ambiente ha determinato l'elenco di tutti i file aggiunti e modificati e subito prima di leggerli. Consente agli autori delle estensioni di riordinare l'elenco dei docname (*al momento*) prima dell'elaborazione, o di aggiungere altri docname che Sphinx non ha considerato come modificati (ma non aggiunge mai docname che non siano in *found_docs*).

Si possono anche rimuovere i nomi dei documenti; farlo con cautela poiché Sphinx considererà i file modificati (??rimossi??) come invariati.

Added in version 1.3.

source-read(*app*, *docname*, *content*)

Parametri

- **app** -- *Sphinx*
- **docname** -- str
- **content** -- list[str] contenente un singolo elemento, che rappresenta il contenuto del file incluso.

Emesso quando un file sorgente è stato letto.

Si può elaborare il **contenuto** e sostituire questo elemento per implementare trasformazioni a livello di sorgente.

Ad esempio, per utilizzare i segni \$ per delimitare le formule matematiche in linea, come in LaTeX, si può utilizzare un'espressione regolare per sostituire $...$ con `:math:`...``.

Added in version 0.5.

include-read(*app*, *relative_path*, *parent_docname*, *content*)

Parametri

- **app** -- *Sphinx*
- **relative_path** -- *Path*⁶⁰³ che rappresenta il file incluso relativo alla *directory sorgente*.

- **parent_docname** -- str del nome del documento che contiene la direttiva `include`⁶⁰⁴.
- **content** -- list[str] contenente un singolo elemento, che rappresenta il contenuto del file incluso.

Emesso quando un file è stato letto con la direttiva `include`⁶⁰⁵.

Si può elaborare il contenuto e sostituire questo elemento per trasformare il contenuto incluso, come con l'evento `source-read`.

Added in version 7.2.5.

Vedi anche:

La direttiva `include`⁶⁰⁶ e l'evento `source-read`.

object-description-transform(*app, domain, objtype, contentnode*)

Parametri

- **app** -- *Sphinx*
- **domain** -- str
- **objtype** -- str
- **contentnode** -- *desc_content*

Emesso quando viene eseguita una direttiva sulla descrizione dell'oggetto. Gli argomenti *domain* e *objtype* sono stringhe che indicano la descrizione dell'oggetto. E *contentnode* è un contenuto per l'oggetto. Può essere modificato sul posto.

Added in version 2.4.

doctree-read(*app, doctree*)

Parametri

- **app** -- *Sphinx*
- **doctree** -- `docutils.nodes.document`

Emesso quando un doctree è stato analizzato e letto dall'ambiente e sta per essere [pickled]. Il doctree può essere modificato sul posto.

missing-reference(*app, env, node, contnode*)

Parametri

- **app** -- *Sphinx*
- **env** -- *BuildEnvironment*

⁶⁰³ <https://docs.python.org/3/library/pathlib.html#pathlib.Path>

⁶⁰⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

⁶⁰⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

⁶⁰⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

- **node** -- Il nodo *pending_xref* da risolvere. I suoi attributi *reftype*, *reftarget*, *modname* e *classname* determinano il tipo e il target del riferimento.
- **contnode** -- Il nodo che trasporta il testo e la formattazione all'interno del riferimento futuro e dovrebbe essere figlio del nodo di riferimento restituito.

restituisce

Un nuovo nodo da inserire nell'albero del documento al posto del nodo, oppure *None* per consentire ad altri gestori di provarlo.

Emesso quando non è possibile risolvere un riferimento incrociato a un oggetto. Se il gestore eventi può risolvere il riferimento, dovrebbe restituire un nuovo nodo *docutils* da inserire nell'albero dei documenti al posto del nodo *node*. Di solito questo nodo è un nodo *reference* contenente *contnode* come figlio. Se il gestore non riesce a risolvere il riferimento incrociato, può restituire *None* per consentire ad altri gestori di provare, oppure sollevare *NoUri* per impedire ad altri gestori di tentare e sopprimere un warning relativo a questo riferimento incrociato irrisolto.

Added in version 0.5.

warn-missing-reference(*app*, *domain*, *node*)

Parametri

- **app** -- *Sphinx*
- **domain** -- Il *Domain* del riferimento mancante.
- **node** -- Il nodo *pending_xref* che non è stato possibile risolvere.

restituisce

True se è stato emesso un warning, altrimenti *None*

Emesso quando un riferimento incrociato a un oggetto non può essere risolto anche dopo *missing-reference*. Se il gestore di eventi può emettere warning per il riferimento mancante, dovrebbe restituire *True*. Le variabili di configurazione *nitpick_ignore* e *nitpick_ignore_regex* impediscono che l'evento venga emesso per i nodi corrispondenti.

Added in version 3.4.

doctree-resolved(*app*, *doctree*, *docname*)

Parametri

- **app** -- *Sphinx*
- **doctree** -- `docutils.nodes.document`
- **docname** -- `str`

Emesso quando un doctree è stato "resolved" dall'ambiente, cioè tutti i riferimenti sono stati risolti e i TOC sono stati inseriti. Il *doctree* può essere modificato sul posto.

Qui è il posto dove sostituire i nodi personalizzati che non hanno metodi visitor nei writer, in modo che non causino errori quando i writer li incontrano.

env-merge-info(*app*, *env*, *docnames*, *other*)

Parametri

- **app** -- *Sphinx*
- **env** -- *BuildEnvironment*
- **docnames** -- list[str]
- **other** -- *BuildEnvironment*

Questo evento viene emesso solo quando è abilitata la lettura parallela dei documenti. Viene emesso una volta per ogni sottoprocesso che ha letto dei documenti.

È necessario gestire questo evento in un'estensione che archivia i dati nell'ambiente in una posizione personalizzata. Altrimenti l'ambiente nel processo main [principale] non sarà a conoscenza delle informazioni archiviate nel sottoprocesso.

other è l'oggetto environment del sub-processo, *env* è l'ambiente del processo principale. *docnames* è un insieme di nomi di documenti che sono stati letti nel sottoprocesso.

Added in version 1.3.

env-updated(*app*, *env*)

Parametri

- **app** -- *Sphinx*
- **env** -- *BuildEnvironment*

restituisce

l'iterabile di str

Emesso dopo aver letto tutti i documenti, quando l'ambiente e tutti i doctree sono ora aggiornati.

Si può restituire un iterabile di docnames dal gestore.er. Tali documenti verranno poi considerati aggiornati, e verranno (ri)scritti in fase di scrittura.

Added in version 0.5.

Cambiato nella versione 1.3: Ora viene utilizzato il valore restituito dagli handler [gestori].

env-get-updated(*app*, *env*)

Parametri

- **app** -- *Sphinx*
- **env** -- *BuildEnvironment*

restituisce

l'iterabile di str

Emesso quando l'ambiente determina quali file sorgenti sono cambiati e devono essere riletti. Si può restituire un iterabile di docname da rileggere.

env-check-consistency(*app*, *env*)

Parametri

- **app** -- *Sphinx*
- **env** -- *BuildEnvironment*

Emesso durante la fase di verifica della Consistency [coerenza]. Si può verificare la coerenza dei metadati per tutti i documenti.

Added in version 1.6.

build-finished(*app*, *exception*)

Parametri

- **app** -- *Sphinx*
- **exception** -- Exception o None

Emesso al termine di una build, prima che Sphinx esca, solitamente utilizzato per la pulizia. Questo evento viene emesso anche quando il processo di build solleva un'eccezione, fornita come argomento di *exception*. L'eccezione viene sollevata nuovamente nell'applicazione dopo l'esecuzione dei gestori eventi. Se il processo di build non ha sollevato eccezioni, *exception* sarà None. Ciò consente di personalizzare le azioni di pulizia in base allo stato dell'eccezione.

Added in version 0.5.

Eventi specifici del builder

Questi eventi vengono emessi da specifici builder.

html-collect-pages(*app*)

Parametri

app -- *Sphinx*

restituisce

iterabile di (*pagename*, *context*, *templatename*) dove *pagename* e *templatename* sono stringhe e *context* è un `dict[str, Any]`.

Emesso quando il builder HTML inizia a scrivere pagine non di documenti.

Si possono aggiungere pagine da scrivere restituendo un iterabile da questo evento.

Added in version 1.0.

html-page-context(*app*, *pagename*, *templatename*, *context*, *doctree*)

Parametri

- **app** -- *Sphinx*

- **pagename** -- str
- **templatename** -- str
- **context** -- dict[str, Any]
- **doctree** -- docutils.nodes.document o None

restituisce

str o None

Emesso quando il builder HTML ha creato un dizionario contestuale con cui eseguire il rendering di un template: può essere utilizzato per aggiungere elementi personalizzati al contesto.

L'argomento *pagename* è il nome canonico della pagina di cui viene eseguito il rendering, ovvero senza il suffisso `.html` e utilizzando gli slash [barre] come separatori di percorso. Il *templatename* è il nome del template da renderizzare, questo sarà `'page.html'` per tutte le pagine dei documenti reST.

L'argomento *context* è un dizionario di valori che vengono forniti al motore del template per eseguire il rendering della pagina e che può essere modificato per includere valori personalizzati.

L'argomento *doctree* sarà un doctree quando la pagina viene creata da un documento reST; sarà None quando la pagina viene creata solo da un template HTML.

Si può restituire una stringa dal gestore, questa sostituirà poi `'page.html'` come template HTML per questa pagina.

Suggerimento: Si possono installare file JS/CSS per la pagina specifica tramite [*Sphinx.add_js_file\(\)*](#) e [*Sphinx.add_css_file\(\)*](#) (dalla versione v3.5.0).

Added in version 0.4.

Cambiato nella versione 1.3: Il valore restituito ora può specificare un nome di un template.

linkcheck-process-uri (*app*, *uri*)

Parametri

- **app** -- [*Sphinx*](#)
- **uri** -- str dell'URI raccolto

restituisce

str o None

Emesso quando il builder linkcheck raccoglie collegamenti ipertestuali dal documento.

Gli event handler possono modificare l'URI restituendo una stringa.

Added in version 4.1.

API del progetto

class sphinx.project.**Project**(*srcdir*: [str](#)⁶⁰⁷ | [os.PathLike](#)⁶⁰⁸[[str](#)⁶⁰⁹], *source_suffix*: [Iterable](#)[[str](#)⁶¹⁰])

Un progetto è l'insieme del codice sorgente dei documenti Sphinx.

discover(*exclude_paths*: [Iterable](#)[[str](#)⁶¹¹] = (), *include_paths*: [Iterable](#)[[str](#)⁶¹²] = ('**',)) → [set](#)⁶¹³[[str](#)⁶¹⁴]

Trovare tutti i file dei documenti nella directory sorgente e inserirli in *docnames*.

doc2path(*docname*: [str](#)⁶¹⁵, *absolute*: [bool](#)⁶¹⁶) → [str](#)⁶¹⁷

Restituisce il nome file per il nome del documento.

Se *absolute* è True, restituisce un percorso assoluto. Altrimenti, ritorna come percorso relativo alla directory sorgente.

path2doc(*filename*: [str](#)⁶¹⁸ | [PathLike](#)⁶¹⁹[[str](#)⁶²⁰]) → [str](#)⁶²¹ | [None](#)⁶²²

Restituisce il docname come nome del file se il file è un documento.

filename deve essere assoluto o relativo alla directory sorgente.

restore(*other*: [Project](#)) → [None](#)⁶²³

Acquisisce un risultato dell'ultima build.

docnames: [set](#)⁶²⁴[[str](#)⁶²⁵]

Il nome dei documenti appartenenti a questo progetto.

source_suffix

source_suffix. Uguale a *source_suffix*.

srcdir

Directory sorgente.

⁶⁰⁷ <https://docs.python.org/3/library/stdtypes.html#str>

⁶⁰⁸ <https://docs.python.org/3/library/os.html#os.PathLike>

⁶⁰⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁶¹⁰ <https://docs.python.org/3/library/stdtypes.html#str>

⁶¹¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁶¹² <https://docs.python.org/3/library/stdtypes.html#str>

⁶¹³ <https://docs.python.org/3/library/stdtypes.html#set>

⁶¹⁴ <https://docs.python.org/3/library/stdtypes.html#str>

⁶¹⁵ <https://docs.python.org/3/library/stdtypes.html#str>

⁶¹⁶ <https://docs.python.org/3/library/functions.html#bool>

⁶¹⁷ <https://docs.python.org/3/library/stdtypes.html#str>

⁶¹⁸ <https://docs.python.org/3/library/stdtypes.html#str>

⁶¹⁹ <https://docs.python.org/3/library/os.html#os.PathLike>

⁶²⁰ <https://docs.python.org/3/library/stdtypes.html#str>

⁶²¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁶²² <https://docs.python.org/3/library/constants.html#None>

⁶²³ <https://docs.python.org/3/library/constants.html#None>

⁶²⁴ <https://docs.python.org/3/library/stdtypes.html#set>

⁶²⁵ <https://docs.python.org/3/library/stdtypes.html#str>

API per la build dell'environment

class sphinx.environment.**BuildEnvironment**

Attributi

app

Riferimento all'oggetto [Sphinx](#) (applicazione).

config

Riferimento all'oggetto [Config](#).

project

Progetto target. Vedere [Project](#).

srcdir

Directory sorgente.

doctreedir

Directory per la memorizzazione di doctree pickled.

events

Un oggetto [EventManager](#).

found_docs

Un insieme di tutti i nomi dei documenti esistenti.

metadata

Dizionario che mappa i nomi dei documenti in "metadata" (consultare [Metadati a livello di file](#)).

titles

Dizionario che mappa i nomi dei documenti ai nodi docutil per il loro titolo principale.

docname

Restituisce il nomedoc del documento attualmente in fase di analisi.

parser

Restituisce il parser utilizzato per analizzare il documento corrente.

Metodi di utilità

doc2path(docname: [str](#)⁶²⁶, base: [bool](#)⁶²⁷ = True) → [str](#)⁶²⁸

Restituisce il nome file per il nome del documento.

Se *base* è True, restituisce il percorso assoluto in self.srcdir. Se *base* è False, restituisce il percorso relativo a self.srcdir.

relfn2path(filename: [str](#)⁶²⁹, docname: [str](#)⁶³⁰ | [None](#)⁶³¹ = None) → [tuple](#)⁶³²[[str](#)⁶³³, [str](#)⁶³⁴]

Restituisce i percorsi a un file a cui fa riferimento un documento, relativo alla radice della documentazione e assoluto.

Nell'input "filename", i nomi di file assoluti sono considerati relativi alla directory sorgente, mentre i nomi di file relativi sono relativi alla directory del documento che li contiene.

note_dependency(filename: [str](#)⁶³⁵) → [None](#)⁶³⁶

Aggiunge *filename* come dipendenza del documento corrente.

Questo significa che il documento verrà re-buildato se questo file cambia.

filename deve essere assoluto o relativo alla directory sorgente.

new_serialno(category: [str](#)⁶³⁷ = "") → [int](#)⁶³⁸

Restituisce un numero di serie, ad es. per target di voci di indice.

È garantito che il numero sia univoco nel documento corrente.

note_reread() → [None](#)⁶³⁹

Aggiunge il documento corrente all'elenco di quelli che verranno riletti automaticamente alla prossima build.

API del Builder

class sphinx.builders.Builder

Questa è la classe base per tutti i builder.

Segue questo workflow [flusso di lavoro] di base:

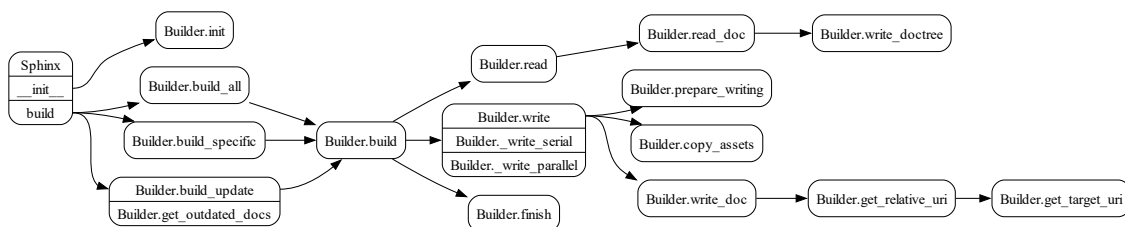


Fig. 2: UML per il workflow della build standard di Sphinx

⁶²⁶ <https://docs.python.org/3/library/stdtypes.html#str>
⁶²⁷ <https://docs.python.org/3/library/functions.html#bool>
⁶²⁸ <https://docs.python.org/3/library/stdtypes.html#str>
⁶²⁹ <https://docs.python.org/3/library/stdtypes.html#str>
⁶³⁰ <https://docs.python.org/3/library/stdtypes.html#str>
⁶³¹ <https://docs.python.org/3/library/constants.html#None>
⁶³² <https://docs.python.org/3/library/stdtypes.html#tuple>
⁶³³ <https://docs.python.org/3/library/stdtypes.html#str>
⁶³⁴ <https://docs.python.org/3/library/stdtypes.html#str>
⁶³⁵ <https://docs.python.org/3/library/stdtypes.html#str>
⁶³⁶ <https://docs.python.org/3/library/constants.html#None>
⁶³⁷ <https://docs.python.org/3/library/stdtypes.html#str>
⁶³⁸ <https://docs.python.org/3/library/functions.html#int>
⁶³⁹ <https://docs.python.org/3/library/constants.html#None>

Attributi Sovrascrivibili [overridable]

Questi attributi dovrebbero essere impostati nelle sottoclassi del builder:

name = ''

Il nome del builder, per l'opzione della riga di comando -b.

format = ''

Il formato di output del builder o '' se non viene prodotto alcun output del documento.

epilog = ''

Il messaggio emesso al completamento con successo della build. Può essere una stringa template in stile printf con le seguenti chiavi: `outdir`, `project`

allow_parallel = False

consente chiamate parallele `write_doc()`

supported_image_types: `list`⁶⁴⁰ [`str`⁶⁴¹] = []

L'elenco dei tipi MIME di formati delle immagini supportate dal builder. I file delle immagini vengono cercati nell'ordine in cui appaiono qui.

supported_remote_images = False

Il builder può produrre documenti di output che, una volta aperti, potrebbero recuperare immagini esterne.

supported_data_uri_images = False

Il formato del file prodotto dal builder consente di includere immagini utilizzando data-URI.

default_translator_class: `type`⁶⁴² [`nodes.NodeVisitor`]

la classe del traduttore di default per il builder. Questo può essere sovrascritto da `set_translator()`.

Metodi Core

Questi metodi sono predefiniti e generalmente non dovrebbero essere sovrascritti [overridden], poiché costituiscono il nucleo del processo di build:

final build_all() → `None`⁶⁴³

Build di tutti i file sorgenti.

final build_specific(*filenames*: `list`⁶⁴⁴ [`str`⁶⁴⁵]) → `None`⁶⁴⁶

Re-build solo quanto necessario per le modifiche nei *filename*.

final build_update() → `None`⁶⁴⁷

Re-build solo ciò che è stato modificato o aggiunto dall'ultima build.

final build(*docnames*: `Iterable` [`str`⁶⁴⁸] | `None`⁶⁴⁹, *summary*: `str`⁶⁵⁰ | `None`⁶⁵¹ = `None`, *method*: `Literal`['all', 'specific', 'update'] = 'update') → `None`⁶⁵²

Metodo principale della build, solitamente chiamato da uno specifico metodo `build_*`.

Prima aggiorna l'ambiente [environment] e poi chiama `write()`.

final read() → `list`⁶⁵³[`str`⁶⁵⁴]

(Ri-)legge tutti i file nuovi o modificati dall'ultimo aggiornamento.

Memorizza tutti i docname dell'ambiente nel formato canonico (ovvero utilizzando SEP come separatore al posto di `os.path.sep`).

final read_doc(docname: str⁶⁵⁵**, *, _cache: bool**⁶⁵⁶ **= True)** → `None`⁶⁵⁷

Analizza un file e aggiunge/aggiorna le voci di inventario per il doctree.

final write_doctree(docname: str⁶⁵⁸**, doctree: document, *, _cache: bool**⁶⁵⁹ **= True)** → `None`⁶⁶⁰

Scriva il doctree in un file, da utilizzare come cache durante le ricompilazioni.

Metodi Sovrascrivibili

Questi devono essere implementati nelle sottoclassi del builder:

get_outdated_docs() → `str`⁶⁶¹ | `Iterable`[`str`⁶⁶²]

Restituisce un iterabile di file di output obsoleti o una stringa che descrive cosa verrà creato da una build di aggiornamento.

Se il builder non genera file individuali corrispondenti ai file sorgenti, restituisce una stringa qui. In tal caso, restituisce un iterabile di quei file che devono essere scritti.

prepare_writing(docnames: set⁶⁶³**[str**⁶⁶⁴**])** → `None`⁶⁶⁵

Un posto dove si può aggiungere logica prima che venga eseguito `write_doc()`

write_doc(docname: str⁶⁶⁶**, doctree: document)** → `None`⁶⁶⁷

Dove effettivamente scrive qualcosa sul filesystem..

get_target_uri(docname: str⁶⁶⁸**, typ: str**⁶⁶⁹ | `None`⁶⁷⁰ = `None`) → `str`⁶⁷¹

Restituisce l'URI target per un nome di documento.

`typ` può essere utilizzato per qualificare la caratteristica del link per i singoli builder.

Questi metodi si possono sovrascrivere nelle sottoclassi del builder:

init() → `None`⁶⁷²

Carica i template necessari ed esegue l'inizializzazione. L'implementazione di default non fa nulla.

write(build_docnames: Iterable⁶⁷³**[str**⁶⁷⁴**] | None**⁶⁷⁵**, updated_docnames: Sequence**⁶⁷⁶**[str**⁶⁷⁷**], method: Literal['all', 'specific', 'update'] = 'update')** → `None`⁶⁷⁸

Scriva i file di output specifici del builder.

copy_assets() → `None`⁶⁷⁹

Dove gli asset (immagini, file statici, ecc.) vengono copiati prima della scrittura

get_relative_uri (*from_*: *str*⁶⁷⁸, *to*: *str*⁶⁷⁹, *typ*: *str*⁶⁸⁰ | *None*⁶⁸¹ = *None*) → *str*⁶⁸²

Restituisce un URI relativo tra due filename sorgenti.

Può sollevare `environment.NoUri` se non c'è modo di restituire un URI sensato.

finish() → *None*⁶⁸³

Completa il processo di build.

L'implementazione di default non fa nulla.

Attributi

Attributi richiamabili dall'istanza del builder:

events

Un oggetto *EventManager*.

Environment Collector API

class sphinx.environment.collectors.EnvironmentCollector

Un EnvironmentCollector è un raccoglitore di dati specifico di ciascun documento.

Raccoglie dati e archivia *BuildEnvironment* come database. Esempi di dati specifici potrebbero essere immagini, file di download, titoli di sezioni, metadati, voci di indice e toctree, ecc.

Nota: Questa classe, essenzialmente, racchiude un sottoinsieme delle *callback degli*

```

640 https://docs.python.org/3/library/stdtypes.html#list
641 https://docs.python.org/3/library/stdtypes.html#str
642 https://docs.python.org/3/library/functions.html#type
643 https://docs.python.org/3/library/constants.html#None
644 https://docs.python.org/3/library/stdtypes.html#list
645 https://docs.python.org/3/library/stdtypes.html#str
646 https://docs.python.org/3/library/constants.html#None
647 https://docs.python.org/3/library/constants.html#None
648 https://docs.python.org/3/library/stdtypes.html#str
649 https://docs.python.org/3/library/constants.html#None
650 https://docs.python.org/3/library/stdtypes.html#str
651 https://docs.python.org/3/library/constants.html#None
652 https://docs.python.org/3/library/constants.html#None
653 https://docs.python.org/3/library/stdtypes.html#list
654 https://docs.python.org/3/library/stdtypes.html#str
655 https://docs.python.org/3/library/stdtypes.html#str
656 https://docs.python.org/3/library/functions.html#bool
657 https://docs.python.org/3/library/constants.html#None
658 https://docs.python.org/3/library/stdtypes.html#str
659 https://docs.python.org/3/library/functions.html#bool
660 https://docs.python.org/3/library/constants.html#None
661 https://docs.python.org/3/library/stdtypes.html#str
662 https://docs.python.org/3/library/stdtypes.html#str
663 https://docs.python.org/3/library/stdtypes.html#set
664 https://docs.python.org/3/library/stdtypes.html#str
665 https://docs.python.org/3/library/constants.html#None
666 https://docs.python.org/3/library/stdtypes.html#str
667 https://docs.python.org/3/library/constants.html#None
668 https://docs.python.org/3/library/stdtypes.html#str
669 https://docs.python.org/3/library/stdtypes.html#str
670 https://docs.python.org/3/library/constants.html#None
671 https://docs.python.org/3/library/stdtypes.html#str
672 https://docs.python.org/3/library/constants.html#None
673 https://docs.python.org/3/library/stdtypes.html#str
674 https://docs.python.org/3/library/constants.html#None
675 https://docs.python.org/3/library/stdtypes.html#str
676 https://docs.python.org/3/library/constants.html#None
677 https://docs.python.org/3/library/constants.html#None
678 https://docs.python.org/3/library/stdtypes.html#str
679 https://docs.python.org/3/library/stdtypes.html#str
680 https://docs.python.org/3/library/stdtypes.html#str
681 https://docs.python.org/3/library/constants.html#None
682 https://docs.python.org/3/library/stdtypes.html#str
683 https://docs.python.org/3/library/constants.html#None

```

eventi Sphinx.

clear_doc(*app*: Sphinx, *env*: BuildEnvironment, *docname*: str⁶⁸⁴) → None⁶⁸⁵

Rimuove i dati specificati di un documento.

Questo metodo è chiamato alla rimozione del documento.

Vedi anche:

env-purge-doc

get_outdated_docs(*app*: Sphinx, *env*: BuildEnvironment, *added*: set⁶⁸⁶[str⁶⁸⁷],
changed: set⁶⁸⁸[str⁶⁸⁹], *removed*: set⁶⁹⁰[str⁶⁹¹]) → list⁶⁹²[str⁶⁹³]

Restituisce un elenco di nomi di documenti da rileggere.

Questo metodo viene chiamato prima di leggere i documenti.

Vedi anche:

env-get-outdated

get_updated_docs(*app*: Sphinx, *env*: BuildEnvironment) → list⁶⁹⁴[str⁶⁹⁵]

Restituisce un elenco di nomi di documenti da rileggere.

Questo metodo viene chiamato dopo aver letto tutti i documenti.

Vedi anche:

env-get-updated

merge_other(*app*: Sphinx, *env*: BuildEnvironment, *docnames*: set⁶⁹⁶[str⁶⁹⁷], *other*:
BuildEnvironment) → None⁶⁹⁸

Unisce i dati specificati riguardanti i nomi dei documenti da un diverso oggetto *BuildEnvironment* che proviene da un sottoprocesso in build parallele.

Vedi anche:

env-merge-info

process_doc(*app*: Sphinx, *doctree*: nodes.document) → None⁶⁹⁹

Elabora un documento e raccoglie dati specifici da esso.

Questo metodo viene chiamato dopo la lettura del documento.

Vedi anche:

doctree-read

API di markup Docutils

Questa sezione descrive l'API per aggiungere elementi di markup reStructuredText (ruoli e direttive).

Ruoli

I ruoli seguono l'interfaccia descritta di seguito. Devono essere registrati da un'estensione che utilizza `Sphinx.add_role()` o `Sphinx.add_role_to_domain()`.

```
def role_function(
    role_name: str, raw_source: str, text: str,
    lineno: int, inliner: Inliner,
    options: dict = {}, content: list = [],
) -> tuple[list[Node], list[system_message]]:
    elements = []
    messages = []
    return elements, messages
```

I parametri *options* e *content* vengono utilizzati solo per i ruoli personalizzati creati tramite la direttiva `role`⁷⁰⁰. Il valore restituito è una tupla di due liste, la prima contenente i nodi di testo e gli elementi del ruolo e la seconda contenente gli eventuali messaggi di sistema generati. Per ulteriori informazioni, consultare la [custom role overview](#)⁷⁰¹ da Docutils.

⁶⁸⁴ <https://docs.python.org/3/library/stdtypes.html#str>
⁶⁸⁵ <https://docs.python.org/3/library/constants.html#None>
⁶⁸⁶ <https://docs.python.org/3/library/stdtypes.html#set>
⁶⁸⁷ <https://docs.python.org/3/library/stdtypes.html#str>
⁶⁸⁸ <https://docs.python.org/3/library/stdtypes.html#set>
⁶⁸⁹ <https://docs.python.org/3/library/stdtypes.html#str>
⁶⁹⁰ <https://docs.python.org/3/library/stdtypes.html#set>
⁶⁹¹ <https://docs.python.org/3/library/stdtypes.html#str>
⁶⁹² <https://docs.python.org/3/library/stdtypes.html#list>
⁶⁹³ <https://docs.python.org/3/library/stdtypes.html#str>
⁶⁹⁴ <https://docs.python.org/3/library/stdtypes.html#list>
⁶⁹⁵ <https://docs.python.org/3/library/stdtypes.html#str>
⁶⁹⁶ <https://docs.python.org/3/library/stdtypes.html#set>
⁶⁹⁷ <https://docs.python.org/3/library/stdtypes.html#str>
⁶⁹⁸ <https://docs.python.org/3/library/constants.html#None>
⁶⁹⁹ <https://docs.python.org/3/library/constants.html#None>
⁷⁰⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#role>
⁷⁰¹ <https://docutils.sourceforge.io/docs/howto/rst-roles.html>

Creazione di ruoli personalizzati

Sphinx fornisce due classi base per la creazione di ruoli personalizzati, *SphinxRole* e *ReferenceRole*.

Queste forniscono un'interfaccia basata su classi per la creazione di ruoli, in cui la logica principale deve essere implementata nel metodo `run()`. Le classi forniscono una serie di metodi e attributi utili, come `self.text`, `self.config` e `self.env`. La classe *ReferenceRole* implementa la logica `title <target>` di Sphinx, esponendo gli attributi `self.target` e `self.title`. Ciò è utile per creare ruoli con riferimenti incrociati.

Direttive

Le direttive sono gestite da classi derivate da `docutils.parsers.rst.Directive`. Devono essere registrati da un'estensione che utilizza *Sphinx.add_directive()* o *Sphinx.add_directive_to_domain()*.

class docutils.parsers.rst.Directive

La sintassi del markup della nuova direttiva è determinata dai seguenti cinque attributi di classe:

required_arguments = 0

Numero di argomenti direttivi richiesti.

optional_arguments = 0

Numero di argomenti facoltativi dopo quelli obbligatori.

final_argument_whitespace = False

L'argomento finale può contenere spazi bianchi?

option_spec = None

Mappatura dei nomi delle opzioni sulle funzioni di validazione.

Le funzioni di validazione delle opzioni accettano un singolo parametro, l'argomento opzione (o `None` se non fornito), e dovrebbero convalidarlo o convertirlo nella forma corretta. Sollevano *ValueError*⁷⁰² o *TypeError*⁷⁰³ per indicare un errore.

Ci sono diversi validatori predefiniti e possibilmente utili nel modulo `docutils.parsers.rst.directives`.

has_content = False

La direttiva può avere un contenuto?

Le nuove direttive devono implementare il metodo *run()*:

run()

Questo metodo deve elaborare gli argomenti, le opzioni e il contenuto della direttiva e restituire un elenco di nodi Docutils/Sphinx che verranno inseriti nell'albero del documento nel punto in cui è stata incontrata la direttiva.

Gli attributi dell'istanza sempre impostati nella direttiva sono:

name

Il nome della direttiva (utile quando si registra la stessa classe di direttive con più nomi).

arguments

Gli argomenti forniti alla direttiva, in forma di lista.

options

Le opzioni fornite alla direttiva, come opzioni di mappatura del dizionario, denominano valori convalidati/convertiti.

content

Il contenuto della direttiva, se fornito, come `ViewList`.

lineno

Il numero di riga assoluto in cui è apparsa la direttiva. Questo non è sempre un valore utile; usare invece `srcline`.

content_offset

Offset interno del contenuto della direttiva. Utilizzato quando si chiama `nested_parse` (vedere di seguito).

block_text

La stringa contenente l'intera direttiva.

state**state_machine**

Lo stato e la macchina a stati che controlla l'analisi. Utilizzato per `nested_parse`.

Vedi anche:

Creazione delle direttive⁷⁰⁴ HOWTO della documentazione di Docutils

Analisi del contenuto della direttiva come `reStructuredText`

Molte direttive conterranno più markup da analizzare. Per fare ciò, si usa una delle seguenti API dal metodo `run()`:

- `SphinxDirective.parse_content_to_nodes()`
- `SphinxDirective.parse_text_to_nodes()`

Il primo metodo analizza tutto il contenuto della direttiva come markup, mentre il secondo analizza solo la stringa `text` specificata. Entrambi i metodi restituiscono i nodi Docutils analizzati in una lista.

I metodi vengono utilizzati come segue:

⁷⁰² <https://docs.python.org/3/library/exceptions.html#ValueError>

⁷⁰³ <https://docs.python.org/3/library/exceptions.html#TypeError>

⁷⁰⁴ <https://docutils.sourceforge.io/docs/howto/rst-directives.html>

```
def run(self) -> list[Node]:
    # either
    parsed = self.parse_content_to_nodes()
    # or
    parsed = self.parse_text_to_nodes('spam spam spam')
    return parsed
```

Nota: I metodi di utilità di cui sopra sono stati aggiunti in Sphinx 7.4. Prima di Sphinx 7.4, per analizzare il contenuto si dovevano utilizzare i seguenti metodi:

- `self.state.nested_parse`
- `sphinx.util.nodes.nested_parse_with_titles()` -- questo consente di avere titoli nel contenuto analizzato.

```
def run(self) -> list[Node]:
    container = docutils.nodes.Element()
    # either
    nested_parse_with_titles(self.state, self.result, container)
    # or
    self.state.nested_parse(self.result, 0, container)
    parsed = container.children
    return parsed
```

Per analizzare i markup in linea, si usa `parse_inline()`. Questo deve essere utilizzato solo per il testo su riga o paragrafo singoli e che non contiene elementi strutturali (titoli, transizioni, direttive, ecc.).

Nota: `sphinx.util.docutils.switch_source_input()` consente di modificare il file sorgente (input) durante l'analisi del contenuto in una direttiva. È utile per analizzare contenuti misti, come in `sphinx.ext.autodoc`, dove viene utilizzato per analizzare le docstring.

```
from sphinx.util.docutils import switch_source_input
from sphinx.util.parsing import nested_parse_to_nodes

# Switch source_input between parsing content.
# Inside this context, all parsing errors and warnings are reported as
# happened in new source_input (in this case, ``self.result``).
with switch_source_input(self.state, self.result):
    parsed = nested_parse_to_nodes(self.state, self.result)
```

Deprecato dalla versione 1.7: Fino a Sphinx 1.6, per questo scopo veniva utilizzato `sphinx.ext.autodoc.AutodocReporter`. Viene sostituito da `switch_source_input()`.

ViewList e StringList

Docutils rappresenta le righe di origine del documento in una classe `StringList`, che eredita da `ViewList`, entrambe nel modulo `docutils.statemachine`. Si tratta di una lista con le funzionalità estese, incluso il fatto che la suddivisione crea visualizzazioni della lista originale e che l'elenco contiene informazioni sui numeri di riga originali.

L'attributo `Directive.content` è una `StringList`. Se si genera contenuto da analizzare come `reStructuredText`, si deve creare una `StringList` per le API Docutils. Le funzioni di utilità fornite da Sphinx lo gestiscono automaticamente. Importanti per la generazione del contenuto sono i seguenti punti:

- Il costruttore di `ViewList` accetta un elenco di stringhe (righe) e un nome sorgente (documento).
- Il metodo `ViewList.append()` accetta anche una riga e un nome sorgente.

API di dominio

class `sphinx.domains.Domain`(*env*: `BuildEnvironment`)

Un dominio è inteso come un gruppo di direttive di descrizione di "object" per oggetti di natura simile e ruoli corrispondenti per creare riferimenti ad essi. Esempi potrebbero essere moduli, classi, funzioni Python, ecc., elementi di un linguaggio di template, ruoli e direttive Sphinx, ecc.

Ogni dominio ha un archivio separato per le informazioni sugli oggetti esistenti e su come fare riferimento ad essi in `self.data`, che deve essere un dizionario. Deve inoltre implementare diverse funzioni che esponano le informazioni sull'oggetto in modo uniforme a parti di Sphinx che consentano all'utente di fare riferimento o cercare oggetti in modo indipendente dal dominio.

Informazioni su `self.data`: poiché tutte le informazioni sugli oggetti e sui riferimenti incrociati sono memorizzate su un'istanza `BuildEnvironment`, anche l'oggetto `domain.data` viene archiviato nel dict `env.domaindata` sotto la chiave `domain.name`. Prima dell'inizio del processo di build, viene istanziato ogni dominio attivo e gli viene assegnato l'oggetto `environment`; il dict `domaindata` deve poi essere inesistente o un dizionario la cui chiave 'version' è uguale all'attributo `data_version` del dominio della classe. Altrimenti viene sollevata l'eccezione `OSError` e l'ambiente serializzato [pickled] viene scartato.

add_object_type(*name*: `str`⁷⁰⁵, *objtype*: `ObjType`) → `None`⁷⁰⁶

Aggiunge un tipo di oggetto.

check_consistency() → `None`⁷⁰⁷

Effettuare controlli di coerenza (**sperimentale**).

clear_doc(*docname*: `str`⁷⁰⁸) → `None`⁷⁰⁹

Rimuovere i trace di un documento negli inventari specifici del dominio.

directive(*name*: *str*⁷¹⁰) → *Callable*⁷¹¹ | *None*⁷¹²

Restituisce una classe "directive adapter" che fornisce sempre alla direttiva registrata il suo nome completo ('domain:name') come `self.name`.

get_enumerable_node_type(*node*: *Node*) → *str*⁷¹³ | *None*⁷¹⁴

Preleva il tipo di nodi enumerabili (sperimentale).

get_full_qualified_name(*node*: *Element*) → *str*⁷¹⁵ | *None*⁷¹⁶

Restituisce il nome completo per il nodo specificato.

get_objects() → *Iterable*[*tuple*⁷¹⁷[*str*⁷¹⁸, *str*⁷¹⁹, *str*⁷²⁰, *str*⁷²¹, *str*⁷²², *int*⁷²³]]

Restituisce un iterabile delle "object descriptions" (*descrizioni degli oggetti*).

Le "object description" sono tuple con sei elementi:

name

Il nome completo.

dispname

Nome da visualizzare durante la ricerca/linking.

type

Tipo di oggetto, una chiave in `self.object_types`.

docname

Il documento in cui si trova.

anchor

Il nome di ancoraggio per l'oggetto.

priority

Quanto è "importante" l'oggetto (determina il posizionamento nei risultati di ricerca). Uno tra:

1

La priorità di default (posizionata prima delle corrispondenze full-text).

0

L'oggetto è importante (posizionato prima degli oggetti con la priorità di default).

2

L'oggetto non è importante (posizionato dopo le corrispondenze full-text).

-1

L'oggetto non dovrebbe apparire affatto nella ricerca.

get_type_name(*type*: *ObjType*, *primary*: *bool*⁷²⁴ = *False*) → *str*⁷²⁵

Restituisce il nome completo per l'*ObjType* specificato.

merge_domaindata(*docnames*: *list*⁷²⁶[*str*⁷²⁷], *otherdata*: *dict*⁷²⁸[*str*⁷²⁹, *Any*⁷³⁰]) → *None*⁷³¹

Mergia [*unisce*] i dati riguardanti i *docname* da un diverso inventario di dati di dominio (provenienti da un sottoprocesso in build parallele).

process_doc(*env*: BuildEnvironment, *docname*: str⁷³², *document*: nodes.document) → None⁷³³

Elabora un documento dopo che è stato letto dall'ambiente

process_field_xref(*pnode*: pending_xref) → None⁷³⁴

Elabora un xref in sospeso creato in un campo doc. Ad esempio, allega informazioni sullo scope corrente.

resolve_any_xref(*env*: BuildEnvironment, *fromdocname*: str⁷³⁵, *builder*: Builder, *target*: str⁷³⁶, *node*: pending_xref, *contnode*: Element) → list⁷³⁷[tuple⁷³⁸[str⁷³⁹, Element]]

Risolve il *node* pending_xref con il *target* specificato.

Il riferimento deriva da un ruolo "qualsiasi" o simile, il che significa che non ne conosciamo il tipo. Altrimenti, gli argomenti sono gli stessi di [resolve_xref\(\)](#).

Il metodo deve restituire una lista (potenzialmente vuota) di tuple ('domain:role', newnode), dove 'domain:role' è il nome di un ruolo che potrebbe aver creato lo stesso riferimento, per esempio 'py:func'. newnode è ciò che restituirà [resolve_xref\(\)](#).

Added in version 1.3.

resolve_xref(*env*: BuildEnvironment, *fromdocname*: str⁷⁴⁰, *builder*: Builder, *typ*: str⁷⁴¹, *target*: str⁷⁴², *node*: pending_xref, *contnode*: Element) → Element | None⁷⁴³

Risolve il *node* pending_xref con il *typ* e il *target* specificati.

Questo metodo dovrebbe restituire un nuovo nodo, per sostituire il nodo xref, contenente il *contnode* che è il contenuto di markup del riferimento incrociato.

Se non è possibile trovare alcuna soluzione, è possibile restituire None; il nodo xref verrà quindi assegnato all'evento [missing-reference](#) e, se ciò non produce alcuna risoluzione, sostituito da *contnode*.

Il metodo può anche sollevare sphinx.environment.NoUri per sopprimere l'emissione dell'evento [missing-reference](#).

role(*name*: str⁷⁴⁴) → RoleFunction | None⁷⁴⁵

Restituisce una funzione adapter di ruolo che fornisce sempre il ruolo registrato col suo nome completo ('domain:name') come primo argomento.

setup() → None⁷⁴⁶

Configura l'oggetto del dominio.

dangling_warnings: dict⁷⁴⁷[str⁷⁴⁸, str⁷⁴⁹] = {}

role name -> un messaggio di warning se manca il riferimento

data: dict⁷⁵⁰

dato valore

data_version = 0

dato sulla versione, lo aumenta quando cambia il formato di *self.data*

directives: `dict`⁷⁵¹[`str`⁷⁵², `type`⁷⁵³[*Directive*]] = {}
directive name -> directive class

enumerable_nodes: `dict`⁷⁵⁴[`type`⁷⁵⁵[*Node*], `tuple`⁷⁵⁶[`str`⁷⁵⁷, *TitleGetter* | *None*⁷⁵⁸]] = {}
node_class -> (enum_node_type, title_getter)

indices: `list`⁷⁵⁹[`type`⁷⁶⁰[*Index*]] = []
una lista di sottoclassi di *Index*

initial_data: `dict`⁷⁶¹ = {}
dati per un environment nuovo

label = ''
domain label: più lunga, più descrittiva (usata nei messaggi)

name = ''
domain name: dovrebbe essere breve, ma unico

object_types: `dict`⁷⁶²[`str`⁷⁶³, *ObjType*] = {}
nome del tipo (solitamente directive) name -> Istanza di *ObjType*

roles: `dict`⁷⁶⁴[`str`⁷⁶⁵, *RoleFunction* | *XRefRole*] = {}
role name -> role callable

<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/typing.html#typing.Callable>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#tuple>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/functions.html#int>
<https://docs.python.org/3/library/functions.html#bool>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#list>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#dict>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/typing.html#typing.Any>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#list>
<https://docs.python.org/3/library/stdtypes.html#tuple>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#dict>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/stdtypes.html#dict>
<https://docs.python.org/3/library/stdtypes.html#dict>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/functions.html#type>
<https://docs.python.org/3/library/stdtypes.html#dict>
<https://docs.python.org/3/library/functions.html#type>
<https://docs.python.org/3/library/stdtypes.html#tuple>
<https://docs.python.org/3/library/stdtypes.html#str>
<https://docs.python.org/3/library/constants.html#None>
<https://docs.python.org/3/library/stdtypes.html#list>
<https://docs.python.org/3/library/functions.html#type>
<https://docs.python.org/3/library/stdtypes.html#dict>
<https://docs.python.org/3/library/stdtypes.html#dict>
<https://docs.python.org/3/library/stdtypes.html#str>

class sphinx.domains.**ObjType**(*lname: str*⁷⁶⁶, **roles: Any*⁷⁶⁷, ***attrs: Any*⁷⁶⁸)

Un `ObjType` è la descrizione di un tipo di oggetto che un dominio può documentare. Nell'attributo `object_types` delle sottoclassi `Domain`, i nomi dei tipi di oggetto vengono mappati alle istanze di questa classe.

Argomenti del costruttore:

- *lname*: nome localizzato del tipo (non include il nome di dominio)
- *roles*: tutti i ruoli che possono fare riferimento ad un oggetto di questo tipo
- *attrs*: attributi dell'oggetto -- attualmente è noto solo "searchprio", che definisce la priorità dell'oggetto nell'indice di ricerca full-text, vedere `Domain.get_objects()`.

class sphinx.domains.**Index**(*domain: Domain*)

Un `Index` è la descrizione di un indice specifico del dominio. Per aggiungere un indice a un dominio, si crea la sottoclasse di `Index`, sovrascrivendo i tre attributi del nome:

- *name* è un identificatore utilizzato per generare nomi di file. Viene utilizzato anche per il target di un collegamento ipertestuale per l'indice. Pertanto, gli utenti possono fare riferimento alla pagina dell'indice utilizzando il ruolo `ref` e una stringa che combina il nome di dominio e l'attributo `name` attribuite (es. `:ref: `py-modindex``).
- *localname* è il titolo della sezione per l'indice.
- *shortname* è un nome breve per l'indice, da utilizzare nella barra delle relazioni nell'output HTML. Può essere vuoto per disabilitare le voci nella barra delle relazioni.

e fornendo un metodo `generate()` method. Quindi, si aggiunge la classe indice alla lista degli indici del dominio. Le estensioni possono aggiungere indici ai domini esistenti utilizzando `add_index_to_domain()`.

Cambiato nella versione 3.0: È possibile fare riferimento alle pagine dell'indice in base al nome del dominio e al nome dell'indice tramite il ruolo `ref`.

abstract `generate`(*docnames: Iterable[str*⁷⁶⁹ | *None*⁷⁷⁰ = *None*) → *tuple*⁷⁷¹[*list*⁷⁷²[*tuple*⁷⁷³[*str*⁷⁷⁴, *list*⁷⁷⁵[`IndexEntry`]]], *bool*⁷⁷⁶]

Preleva le voci dell'indice.

Se viene fornito `docnames`, si limita alle voci che fanno riferimento a questi `docname`.

Il valore restituito è una tupla di (`content`, `collapse`):

collapse

Un valore booleano che determina se le voci secondarie devono iniziare compresse (per i formati di output che supportano la compressione delle voci secondarie).

⁷⁶⁶ <https://docs.python.org/3/library/stdtypes.html#str>

⁷⁶⁷ <https://docs.python.org/3/library/typing.html#typing.Any>

⁷⁶⁸ <https://docs.python.org/3/library/typing.html#typing.Any>

content:

Una sequenza di tuple (*letter*, *entries*), dove *letter* è lo "heading" per le *entries* date, solitamente la lettera iniziale, e *entries* è una sequenza di singole voci. Ogni voce è una sequenza [*name*, *subtype*, *docname*, *anchor*, *extra*, *qualifier*, *descr*]. Le voci in questa sequenza hanno il seguente significato:

name

Il nome della voce di indice da visualizzare.

subtype

Il tipo correlato alla voce secondaria. Uno tra:

0

Una voce normale.

1

Una voce con sottovoci.

2

Una sottovoce.

docname

docname dove si trova la voce.

anchor

Ancoraggio per la voce all'interno di *docname*

extra

Informazioni extra per la voce.

qualifier

Qualificatore per la descrizione.

descr

Descrizione per la voce.

Il qualificatore e la descrizione non vengono visualizzati per alcuni formati di output come LaTeX.

class `sphinx.directives.ObjectDescription`(*name*, *arguments*, *options*, *content*,
lineno, *content_offset*, *block_text*,
state, *state_machine*)

Direttiva per descrivere una classe, funzione o oggetto simile. Non utilizzato direttamente, ma sottoclassato (nelle direttive specifiche del dominio) per aggiungere un comportamento personalizzato.

⁷⁶⁹ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁷⁰ <https://docs.python.org/3/library/constants.html#None>
⁷⁷¹ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁷⁷² <https://docs.python.org/3/library/stdtypes.html#list>
⁷⁷³ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁷⁷⁴ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁷⁵ <https://docs.python.org/3/library/stdtypes.html#list>
⁷⁷⁶ <https://docs.python.org/3/library/functions.html#bool>

_object_hierarchy_parts(*sig_node*: *desc_signature*) → tuple⁷⁷⁷[str⁷⁷⁸, ...]

Restituisce una tupla di stringhe, una voce per ciascuna parte della gerarchia dell'oggetto (ad esempio ('module', 'submodule', 'Class', 'method')). La tupla restituita viene utilizzata per nidificare correttamente i figli all'interno dei genitori nel sommario e può anche essere utilizzata all'interno del metodo `_toc_entry_name()`.

Questo metodo non deve essere utilizzato senza la generazione del sommario.

_toc_entry_name(*sig_node*: *desc_signature*) → str⁷⁷⁹

Restituisce il testo della voce del sommario per l'oggetto.

Questa funzione viene chiamata una volta, in `run()`, per impostare il nome per la voce del sommario (un attributo speciale `_toc_name` è impostato sul nodo oggetto, successivamente utilizzato in `environment.collectors.tocTree.TocTreeCollector.process_doc().build_toc()` quando vengono raccolte le voci del sommario).

Per supportare le voci del sommario per i propri oggetti, i domini devono sovrascrivere questo metodo, rispettando anche l'impostazione di configurazione `toc_object_entries_show_parents`. I domini devono inoltre sovrascrivere `_object_hierarchy_parts()`, con una voce (stringa) per ciascuna parte della gerarchia dell'oggetto. Il risultato di questo metodo viene impostato sul nodo firma ed è possibile accedervi come `sig_node['_toc_parts']` per l'utilizzo all'interno di questo metodo. La tupla risultante viene utilizzata anche per nidificare correttamente i figli all'interno dei genitori nel sommario.

Un esempio di implementazione di questo metodo è all'interno del dominio Python (`PyObject._toc_entry_name()`). Il dominio python imposta l'attributo `_toc_parts` all'interno del metodo `handle_signature()`.

add_target_and_index(*name*: *ObjDescT*, *sig*: str⁷⁸⁰, *signode*: *desc_signature*) → None⁷⁸¹

Aggiunge ID e voci di riferimento incrociato a `self.indexnode`, se applicabile.

name è qualunque cosa restituita da `handle_signature()`.

after_content() → None⁷⁸²

Chiamato dopo aver analizzato il contenuto. Utilizzato per resettare le informazioni sul contesto della direttiva corrente nell'ambiente di build.

before_content() → None⁷⁸³

Chiamato prima di analizzare il contenuto. Utilizzato per impostare informazioni sul contesto della direttiva corrente nell'ambiente di build.

get_signatures() → list⁷⁸⁴[str⁷⁸⁵]

Recupera le signature (*firme*) per documentare dagli argomenti della direttiva. Per default, le firme vengono fornite come argomenti, una per riga.

handle_signature(*sig*: str⁷⁸⁶, *signode*: *desc_signature*) → *ObjDescT*

Analizza la firma *sig* in singoli nodi e li aggiunge a *into signode*. Se viene sollevata

ValueError, l'analisi viene interrotta e l'intero *sig* viene inserito in un singolo nodo `desc_name`.

Il valore restituito dovrebbe essere un valore che identifica l'oggetto. Viene passato a `add_target_and_index()` senza modifiche e altrimenti utilizzato solo per saltare i duplicati.

run() → `list`⁷⁸⁷[Node]

Funzione di immissione della direttiva principale, richiamata dai docutils quando incontrano la direttiva.

Questa direttiva è pensata per essere facilmente suddivisa in sottoclassi, quindi delega a diversi metodi aggiuntivi. Cosa fa:

- scopre se chiamata come direttiva specifica del dominio, setta `self.domain`
- crea un nodo *desc* per adattare tutta la descrizione all'interno
- analizza le opzioni standard, attualmente *no-index*
- crea un nodo indice se necessario come `self.indexnode`
- analizza tutte le firme fornite (come restituite da `self.get_signatures()`) utilizzando `self.handle_signature()`, che dovrebbe restituire un nome o sollevare `ValueError`
- aggiunge voci di indice usando `self.add_target_and_index()`
- analizza il contenuto e gestisce i campi doc al suo interno

transform_content(*content_node*: `desc_content`) → `None`⁷⁸⁸

Chiamata dopo aver creato il contenuto tramite l'analisi nidificata, ma prima che venga emesso l'evento `object-description-transform` e prima che gli info-fields vengano trasformati. Può essere utilizzata per manipolare il contenuto.

final_argument_whitespace = `True`

L'argomento finale può contenere spazi bianchi?

has_content = `True`

La direttiva può avere un contenuto?

```
option_spec: ClassVar789[dict790[str791, Callable792[[str793], Any794]]]
= {'no-contents-entry': <function flag>, 'no-index': <function
flag>, 'no-index-entry': <function flag>, 'no-typesetting':
<function flag>, 'nocontentsentry': <function flag>, 'noindex':
<function flag>, 'noindexentry': <function flag>}
```

Mappatura dei nomi delle opzioni sulle funzioni di validazione.

optional_arguments = `0`

Numero di argomenti facoltativi dopo quelli obbligatori.

required_arguments = `1`

Numero di argomenti direttivi richiesti.

Dominio Python

class sphinx.domains.python.**PythonDomain**(*env*: BuildEnvironment)

Dominio del linguaggio Python.

objects

modules

note_object(*name*: [str](#)⁷⁹⁵, *objtype*: [str](#)⁷⁹⁶, *node_id*: [str](#)⁷⁹⁷, *aliased*: [bool](#)⁷⁹⁸ = False, *location*: [Any](#)⁷⁹⁹ = None) → [None](#)⁸⁰⁰

Nota un oggetto Python per i riferimenti incrociati.

Added in version 2.1.

note_module(*name*: [str](#)⁸⁰¹, *node_id*: [str](#)⁸⁰², *synopsis*: [str](#)⁸⁰³, *platform*: [str](#)⁸⁰⁴, *deprecated*: [bool](#)⁸⁰⁵) → [None](#)⁸⁰⁶

Nota un modulo Python per i riferimenti incrociati.

Added in version 2.1.

⁷⁷⁷ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁷⁷⁸ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁷⁹ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁸⁰ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁸¹ <https://docs.python.org/3/library/constants.html#None>
⁷⁸² <https://docs.python.org/3/library/constants.html#None>
⁷⁸³ <https://docs.python.org/3/library/constants.html#None>
⁷⁸⁴ <https://docs.python.org/3/library/stdtypes.html#list>
⁷⁸⁵ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁸⁶ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁸⁷ <https://docs.python.org/3/library/stdtypes.html#list>
⁷⁸⁸ <https://docs.python.org/3/library/constants.html#None>
⁷⁸⁹ <https://docs.python.org/3/library/typing.html#typing.ClassVar>
⁷⁹⁰ <https://docs.python.org/3/library/stdtypes.html#dict>
⁷⁹¹ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁹² <https://docs.python.org/3/library/typing.html#typing.Callable>
⁷⁹³ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁹⁴ <https://docs.python.org/3/library/typing.html#typing.Any>
⁷⁹⁵ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁹⁶ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁹⁷ <https://docs.python.org/3/library/stdtypes.html#str>
⁷⁹⁸ <https://docs.python.org/3/library/functions.html#bool>
⁷⁹⁹ <https://docs.python.org/3/library/typing.html#typing.Any>
⁸⁰⁰ <https://docs.python.org/3/library/constants.html#None>
⁸⁰¹ <https://docs.python.org/3/library/stdtypes.html#str>
⁸⁰² <https://docs.python.org/3/library/stdtypes.html#str>
⁸⁰³ <https://docs.python.org/3/library/stdtypes.html#str>
⁸⁰⁴ <https://docs.python.org/3/library/stdtypes.html#str>
⁸⁰⁵ <https://docs.python.org/3/library/functions.html#bool>
⁸⁰⁶ <https://docs.python.org/3/library/constants.html#None>

API del Parser

La documentazione di docutils [descrive](#)⁸⁰⁷ i parser come segue:

Il Parser analizza il documento di input e crea una rappresentazione dell'albero dei nodi.

In Sphinx, i moduli parser funzionano come i docutils. I parser sono registrati su Sphinx tramite estensioni che utilizzano le "Application API"; `Sphinx.add_source_suffix()` e `Sphinx.add_source_parser()`.

Il *source suffix* è una mappatura dal suffisso del file al tipo di file. Per esempio, il file `.rst` è mappato al tipo `'restructuredtext'`. Sphinx utilizza il tipo di file per cercare parser dall'elenco registrato. Durante la ricerca, Sphinx fa riferimento all'attributo `Parser.supported` e seleziona un parser che contiene il tipo di file nell'attributo.

Gli utenti possono sovrascrivere le mappature dei suffissi dei sorgenti utilizzando *source_suffix* come segue:

```
# a mapping from file suffix to file types
source_suffix = {
    '.rst': 'restructuredtext',
    '.md': 'markdown',
}
```

Si devono indicare i tipi di file supportati dal parser. Ciò consentirà agli utenti di configurare le proprie impostazioni in modo appropriato.

class sphinx.parsers.Parser

Una classe base di parser di sorgenti. I parser aggiuntivi dovrebbero ereditare questa classe invece di `docutils.parsers.Parser`. Rispetto a `docutils.parsers.Parser`, questa classe migliora l'accessibilità alle API Sphinx.

Le sottoclassi possono accedere agli oggetti 'core' del runtime di sphinx (app, config e env).

set_application(app: [Sphinx](#)) → [None](#)⁸⁰⁸

set_application verrà chiamato da Sphinx per impostare app e altre variabili di istanza

Parametri

app ([sphinx.application.Sphinx](#)) -- Oggetto applicazione Sphinx

config: [Config](#)

L'oggetto config

env: [BuildEnvironment](#)

L'oggetto environment

⁸⁰⁷ <https://docutils.sourceforge.io/docs/dev/hacking.html#parsing-the-document>

⁸⁰⁸ <https://docs.python.org/3/library/constants.html#None>

Classi dei nodi Doctree aggiunte da Sphinx

Nodi per descrizioni di oggetti specifici del dominio

Nodi di primo livello

Questi nodi costituiscono i livelli più alti delle descrizioni degli oggetti.

class sphinx.addnodes.desc(*rawsource*="", **children*, ***attributes*)

Nodo per una lista di firme di oggetti e una loro descrizione comune.

Contiene uno o più nodi *desc_signature* e poi un singolo nodo *desc_content*.

Questo nodo ha sempre due classi:

- Il nome del dominio a cui appartiene, ad esempio `py` o `cpp`.
- Il nome del tipo di oggetto nel dominio, ad esempio `function`.

class sphinx.addnodes.desc_signature(**args*: *Any*⁸⁰⁹, ***kwargs*: *Any*⁸¹⁰)

Nodo per la firma di un singolo oggetto.

Per default, la firma è su una riga singola. Impostare `is_multiline = True` per descrivere una firma su più righe. In tal caso tutti i nodi figli devono essere nodi *desc_signature_line*.

Questo nodo ha sempre le classi `sig`, `sig-object` e il dominio a cui appartiene.

class sphinx.addnodes.desc_signature_line(*rawsource*="", *text*="", **children*, ***attributes*)

Nodo per una riga nella firma di un oggetto su più righe.

Dovrebbe essere utilizzato solo come figlio di una *desc_signature* con `is_multiline` settato a `True`. Settare `add_permalink = True` per la riga che dovrebbe ottenere il permalink.

class sphinx.addnodes.desc_content(*rawsource*="", **children*, ***attributes*)

Nodo per il contenuto della descrizione dell'oggetto.

Deve essere l'ultimo nodo figlio in un nodo *desc*.

class sphinx.addnodes.desc_inline(*domain*: *str*⁸¹¹, **args*: *Any*⁸¹², ***kwargs*: *Any*⁸¹³)

Nodo per un frammento di firma nel testo in linea.

Questo viene utilizzato ad esempio per ruoli come `cpp:expr`.

Questo nodo ha sempre le classi `sig`, `sig-inline` e il nome del dominio a cui appartiene.

⁸⁰⁹ <https://docs.python.org/3/library/typing.html#typing.Any>

⁸¹⁰ <https://docs.python.org/3/library/typing.html#typing.Any>

⁸¹¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁸¹² <https://docs.python.org/3/library/typing.html#typing.Any>

⁸¹³ <https://docs.python.org/3/library/typing.html#typing.Any>

Nodi per la struttura di alto livello nelle firme

Questi nodi capitano nei nodi `desc_signature` non multilinea e nei nodi `desc_signature_line`.

class sphinx.addnodes.**desc_name**(*args: *Any*⁸¹⁴, **kwargs: *Any*⁸¹⁵)

Nodo per il nome dell'oggetto principale.

Per esempio, nella dichiarazione di una classe Python `MyModule.MyClass`, il nome principale è `MyClass`.

Questo nodo ha sempre la classe `sig-name`.

class sphinx.addnodes.**desc_addname**(*args: *Any*⁸¹⁶, **kwargs: *Any*⁸¹⁷)

Nodo per parti di nome aggiuntive per un oggetto.

Ad esempio, nella dichiarazione di una classe Python `MyModule.MyClass`, la parte aggiuntiva del nome è `MyModule..`

Questo nodo ha sempre la classe `sig-prename`.

class sphinx.addnodes.**desc_type**(rawsource="", text="", *children, **attributes)

Nodo per i tipi restituiti o i nomi dei tipi di oggetto.

class sphinx.addnodes.**desc_returns**(rawsource="", text="", *children, **attributes)

Nodo per un'annotazione "returns" (a la `->` in Python).

class sphinx.addnodes.**desc_parameterlist**(rawsource="", text="", *children, **attributes)

Nodo per un elenco di parametri generali.

Per default, l'elenco dei parametri è scritto in linea con il resto della firma. Impostare `multi_line_parameter_list = True` per descrivere un elenco di parametri su più righe. In tal caso ciascun parametro verrà scritto su una propria riga indentata.

class sphinx.addnodes.**desc_parameter**(rawsource="", text="", *children, **attributes)

Nodo per un singolo parametro.

class sphinx.addnodes.**desc_optional**(rawsource="", text="", *children, **attributes)

Nodo per contrassegnare parti opzionali della lista di parametri.

class sphinx.addnodes.**desc_annotation**(rawsource="", text="", *children, **attributes)

Nodo per annotazioni della firma (non annotazioni in stile Python 3).

⁸¹⁴ <https://docs.python.org/3/library/typing.html#typing.Any>

⁸¹⁵ <https://docs.python.org/3/library/typing.html#typing.Any>

⁸¹⁶ <https://docs.python.org/3/library/typing.html#typing.Any>

⁸¹⁷ <https://docs.python.org/3/library/typing.html#typing.Any>

Nodi per elementi di testo della firma

Questi nodi ereditano `desc_sig_element` e sono generalmente tradotti in `docutils.nodes.inline` da `SigElementFallbackTransform`.

Le estensioni possono creare nodi aggiuntivi simili a `desc_sig_*` ma affinché `SigElementFallbackTransform` li traduca automaticamente in nodi inline, devono essere aggiunti a `SIG_ELEMENTS` tramite l'argomento della parola chiave `_sig_element=True` di `desc_sig_element`, per es.:

```
class desc_custom_sig_node(desc_sig_element, _sig_
    element=True): ...
```

Per compatibilità con le versioni precedenti, è ancora possibile aggiungere i nodi direttamente utilizzando `SIG_ELEMENTS.add(desc_custom_sig_node)`.

`sphinx.addnodes.SIG_ELEMENTS: set818[type819[desc_sig_element]]`

Un insieme di classi che eritano `desc_sig_element`. Si prevede che ciascuna classe del nodo venga gestita dalla classe translator del builder se quest'ultima non eredita da `SphinxTranslator`.

Questo set può essere esteso manualmente da estensioni di terze parti o creando una sottoclasse di `desc_sig_element` e utilizzando l'argomento della parola chiave `_sig_element=True`.

```
class sphinx.addnodes.desc_sig_element(rawsource: str820 = "", text: str821 = "",
    *children: Element, **attributes: Any)
```

Classe genitore comune di nodi per il testo in linea di una firma.

```
class sphinx.addnodes.desc_sig_space(rawsource: str822 = "", text: str823 = '',
    *children: Element, **attributes: Any)
```

Nodo per uno spazio in una firma.

```
class sphinx.addnodes.desc_sig_name(rawsource: str824 = "", text: str825 = "",
    *children: Element, **attributes: Any)
```

Nodo per un identificatore in una firma.

```
class sphinx.addnodes.desc_sig_operator(rawsource: str826 = "", text: str827 = "",
    *children: Element, **attributes: Any)
```

Nodo per un operatore in una firma.

⁸¹⁸ <https://docs.python.org/3/library/stdtypes.html#set>

⁸¹⁹ <https://docs.python.org/3/library/functions.html#type>

⁸²⁰ <https://docs.python.org/3/library/stdtypes.html#str>

⁸²¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁸²² <https://docs.python.org/3/library/stdtypes.html#str>

⁸²³ <https://docs.python.org/3/library/stdtypes.html#str>

⁸²⁴ <https://docs.python.org/3/library/stdtypes.html#str>

⁸²⁵ <https://docs.python.org/3/library/stdtypes.html#str>

⁸²⁶ <https://docs.python.org/3/library/stdtypes.html#str>

⁸²⁷ <https://docs.python.org/3/library/stdtypes.html#str>

```
class sphinx.addnodes.desc_sig_punctuation(rawsource: str828 = "", text: str829 = "",
                                           *children: Element, **attributes: Any)
```

Nodo per la punteggiatura in una firma.

```
class sphinx.addnodes.desc_sig_keyword(rawsource: str830 = "", text: str831 = "",
                                       *children: Element, **attributes: Any)
```

Nodo per una parola chiave generale in una firma.

```
class sphinx.addnodes.desc_sig_keyword_type(rawsource: str832 = "", text: str833 = "",
                                           *children: Element, **attributes: Any)
```

Nodo per una parola chiave che è un tipo nativo in una firma.

```
class sphinx.addnodes.desc_sig_literal_number(rawsource: str834 = "", text: str835 = "",
                                             *children: Element,
                                             **attributes: Any)
```

Nodo per un valore letterale numerico in una firma.

```
class sphinx.addnodes.desc_sig_literal_string(rawsource: str836 = "", text: str837 = "",
                                             *children: Element,
                                             **attributes: Any)
```

Nodo per una stringa letterale in una firma.

```
class sphinx.addnodes.desc_sig_literal_char(rawsource: str838 = "", text: str839 = "",
                                           *children: Element, **attributes: Any)
```

Nodo per un carattere letterale in una firma.

Nuovi costrutti simili ad ammonizioni

```
class sphinx.addnodes.versionmodified(rawsource="", text="", *children,
                                       **attributes)
```

Nodo per le voci di modifica della versione.

Attualmente utilizzato per le direttive "versionadded", "versionchanged", "deprecated" e "versionremoved".

⁸²⁸ <https://docs.python.org/3/library/stdtypes.html#str>

⁸²⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁸³⁰ <https://docs.python.org/3/library/stdtypes.html#str>

⁸³¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁸³² <https://docs.python.org/3/library/stdtypes.html#str>

⁸³³ <https://docs.python.org/3/library/stdtypes.html#str>

⁸³⁴ <https://docs.python.org/3/library/stdtypes.html#str>

⁸³⁵ <https://docs.python.org/3/library/stdtypes.html#str>

⁸³⁶ <https://docs.python.org/3/library/stdtypes.html#str>

⁸³⁷ <https://docs.python.org/3/library/stdtypes.html#str>

⁸³⁸ <https://docs.python.org/3/library/stdtypes.html#str>

⁸³⁹ <https://docs.python.org/3/library/stdtypes.html#str>

class sphinx.addnodes.seealso(rawsource="", *children, **attributes)
Ammonizione personalizzata "see also".

Altri nodi a livello di paragrafo

class sphinx.addnodes.compact_paragraph(rawsource="", text="", *children, **attributes)

Nodo per un paragrafo compatto (che non fa mai un nodo <p>).

Nuovi nodi in linea

class sphinx.addnodes.index(rawsource="", text="", *children, **attributes)

Nodo per le voci dell'indice.

Questo nodo viene creato dalla direttiva `index` e ha un attributo, `entries`. Il suo valore è un elenco di 5 tuple di (`entrytype`, `entryname`, `target`, `ignored`, `key`).

`entrytype` è uno tra "single", "pair", "double", "triple".

`key` sono i caratteri di categorizzazione (solitamente un singolo carattere) per la pagina dell'indice generale. Per i dettagli, vedere anche: [glossary](#) e l'issue #2320.

class sphinx.addnodes.pending_xref(rawsource="", *children, **attributes)

Nodo per i riferimenti incrociati che non possono essere risolti senza informazioni complete su tutti i documenti.

Questi nodi vengono risolti prima di scrivere l'output, in `BuildEnvironment.resolve_references`.

class sphinx.addnodes.pending_xref_condition(rawsource="", text="", *children, **attributes)

Nodo che rappresenta un potenziale modo per creare un riferimento incrociato e la condizione in cui questo modo dovrebbe essere utilizzato.

Questo nodo può essere posizionato solo sotto un nodo `pending_xref`. Un nodo **pending_xref** non deve contenere nodi `pending_xref_condition` oppure deve contenere solo nodi `pending_xref_condition`.

Il risolutore di riferimenti incrociati sostituirà un `pending_xref` che contiene nodi `pending_xref_condition` con il contenuto di esattamente uno di questi nodi **pending_xref_condition**. Utilizza l'attributo `condition` per decidere quale contenuto del nodo `pending_xref_condition` usare. Consideriamo ad esempio come agisce il risolutore di riferimenti incrociati su:

```
<pending_xref refdomain="py" reftarget="io.StringIO ...>
  <pending_xref_condition condition="resolved">
    <literal>
      StringIO
    <pending_xref_condition condition="*">
```

(continues on next page)

(continua dalla pagina precedente)

```
<literal>
    io.StringIO
```

Se il risolutore di riferimenti incrociati risolve correttamente il riferimento incrociato, riscrive **pending_xref** come:

```
<reference>
    <literal>
        StringIO
```

Altrimenti, se la risoluzione del riferimento incrociato fallisce, riscrive **pending_xref** come:

```
<reference>
    <literal>
        io.StringIO
```

Il nodo **pending_xref_condition** dovrebbe avere l'attributo **condition**. I domini possono essere memorizzati nelle loro condizioni individuali nell'attributo per filtrare i contenuti durante la fase di risoluzione. Come nome riservato di condizione, `condition=""` viene utilizzato per il fallback in caso di errore di risoluzione. Inoltre, come nome di condizione consigliato, `condition="resolved"` rappresenta una risoluzione riuscita nel modulo intersphinx.

Added in version 4.0.

```
class sphinx.addnodes.literal_emphasis(rawsource="", text="", *children,
                                       **attributes)
```

Nodo che si comporta come *emphasis*, ma non vengono applicati ulteriori elaboratori di testo (ad esempio *smartypants* per l'output HTML).

```
class sphinx.addnodes.download_reference(rawsource="", text="", *children,
                                         **attributes)
```

Nodo per i riferimenti di download, simile a `pending_xref`.

Nodi speciali

```
class sphinx.addnodes.only(rawsource="", *children, **attributes)
```

Nodo per direttive "only" (inclusione condizionale basata sui tag).

```
class sphinx.addnodes.highlightlang(rawsource="", *children, **attributes)
```

Inserito per impostare le opzioni del linguaggio di evidenziazione e del numero di riga per i blocchi di codice successivi.

Non dovrebbe essere necessario generare i nodi seguenti nelle estensioni.

```
class sphinx.addnodes.glossary(rawsource="", *children, **attributes)
```

Nodo per inserire un glossario.

class sphinx.addnodes.toctree(rawsource="", *children, **attributes)

Nodo per l'inserimento di un "TOC tree".

class sphinx.addnodes.start_of_file(rawsource="", *children, **attributes)

Nodo per contrassegnare l'inizio di un nuovo file, utilizzato solo nel builder LaTeX.

class sphinx.addnodes.productionlist(rawsource="", *children, **attributes)

Nodo per le liste di produzione grammaticale.

Contiene nodi production.

class sphinx.addnodes.production(rawsource="", text="", *children, **attributes)

Nodi per una singola regola grammaticale.

API per i Log

sphinx.util.logging.getLogger(name)

Ottiene il logger racchiuso da [sphinx.util.logging.SphinxLoggerAdapter](#).

Il logger Sphinx utilizza sempre il namespace `sphinx.*` per essere indipendente dalle impostazioni del logger root. Garantisce che il logging sia coerente anche se un'estensione di terze parti o un'applicazione importata ne resetta le impostazioni.

Esempio d'uso:

```
>>> from sphinx.util import logging
>>> logger = logging.getLogger(__name__)
>>> logger.info('Hello, this is an extension!')
Hello, this is an extension!
```

class sphinx.util.logging.SphinxLoggerAdapter(logging.LoggerAdapter)

LoggerAdapter che consente parole chiave type e subtype.

error(msg, *args, **kwargs)

critical(msg, *args, **kwargs)

warning(msg, *args, **kwargs)

Inserisce un messaggio su questo logger con il livello specificato. Fondamentalmente, gli argomenti sono gli stessi del modulo logging di Python.

Inoltre, il logger Sphinx supporta i seguenti argomenti di parole chiave:

type, *subtype*

Categorie di log per i warning. Viene utilizzato per sopprimere i warning tramite l'impostazione [suppress_warnings](#).

location

Dove è avvenuto il warning. Viene utilizzato per includere il path e il numero di riga in ciascun log. Consente docname, tuple di docname e numeri di riga e nodi:


```

logger = sphinx.util.logging.getLogger(__name__)
logger.warning('Warning happened!', location='index')
logger.warning('Warning happened!', location=('chapter1/
→index', 10))
logger.warning('Warning happened!', location=some_node)

```

color

Il colore dei log. Per default, i log a livello di errore sono colorati in "darkred" (rosso scuro), quelli a livello critico non sono colorati e quelli a livello di warning sono in "red" (rosso).

log(*level*, *msg*, **args*, ***kwargs*)

info(*msg*, **args*, ***kwargs*)

verbose(*msg*, **args*, ***kwargs*)

debug(*msg*, **args*, ***kwargs*)

Inserisce un messaggio su questo logger con il livello specificato. Fondamentalmente, gli argomenti sono gli stessi del modulo logging di Python.

Inoltre, il logger Sphinx supporta i seguenti argomenti di parole chiave:

nonl

Se true, il logger non "piega" (manda a capo) le righe alla fine del messaggio di registro. Il default è False.

location

Dove è stato emesso il messaggio. Per maggiori dettagli, vedere [*SphinxLoggerAdapter.warning\(\)*](#).

color

Il colore dei log. Per default, i log di livello "info" e "verbose" non sono colorati, mentre quelli di livello debug in "darkgray" (grigio scuro).

sphinx.util.logging.pending_logging()

Il gestore del contesto per rinviare temporaneamente tutti i log.

Per esempio:

```

>>> with pending_logging():
>>>     logger.warning('Warning message!') # not flushed yet
>>>     some_long_process()
>>>
Warning message! # the warning is flushed here

```

sphinx.util.logging.pending_warnings()

Il gestore del contesto per rinviare temporaneamente tutti i log warning.

Simile a [*pending_logging\(\)*](#).

`sphinx.util.logging.prefixed_warnings()`

Il gestore del contesto per anteporre temporaneamente il prefisso a tutti i log warning.

Per esempio:

```
>>> with prefixed_warnings("prefix:"):
>>>     logger.warning('Warning message!') # => prefix: Warning_
      ↳message!
```

Added in version 2.0.

API i18n

`sphinx.locale.init(locale_dirs: Iterable[str840 | None841], language: str842 | None843, catalog: str844 = 'sphinx', namespace: str845 = 'general') → tuple846[NullTranslations, bool847]`

Cerca i cataloghi dei messaggi in *locale_dirs* e si assicura che ci sia almeno un catalogo NullTranslations impostato in *translators*. Se richiamato più volte o se vengono trovati diversi file .mo, i loro contenuti vengono fusi insieme (rendendo così `init` rientrante).

`sphinx.locale.init_console(locale_dir: str848 | None849 = None, catalog: str850 = 'sphinx') → tuple851[NullTranslations852, bool853]`

Inizializza [*locale*] per la console.

Added in version 1.8.

`sphinx.locale.get_translation(catalog: str854, namespace: str855 = 'general') → Callable[[str856], str857]`

Ottiene una funzione di traduzione basata su *catalog* e *namespace*.

L'estensione può utilizzare questa API per tradurre i messaggi sull'estensione:

```
import os
from sphinx.locale import get_translation

MESSAGE_CATALOG_NAME = 'myextension' # name of *.pot, *.po and *.mo
```

(continues on next page)

⁸⁴⁰ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁴¹ <https://docs.python.org/3/library/constants.html#None>

⁸⁴² <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁴³ <https://docs.python.org/3/library/constants.html#None>

⁸⁴⁴ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁴⁵ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁴⁶ <https://docs.python.org/3/library/stdtypes.html#tuple>

⁸⁴⁷ <https://docs.python.org/3/library/functions.html#bool>

⁸⁴⁸ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁴⁹ <https://docs.python.org/3/library/constants.html#None>

⁸⁵⁰ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁵¹ <https://docs.python.org/3/library/stdtypes.html#tuple>

⁸⁵² <https://docs.python.org/3/library/gettext.html#gettext.NullTranslations>

⁸⁵³ <https://docs.python.org/3/library/functions.html#bool>

(continua dalla pagina precedente)

```

→mo files
_ = get_translation(MESSAGE_CATALOG_NAME)
text = _('Hello Sphinx!')

def setup(app):
    package_dir = os.path.abspath(os.path.dirname(__file__))
    locale_dir = os.path.join(package_dir, 'locales')
    app.add_message_catalog(MESSAGE_CATALOG_NAME, locale_dir)

```

Con questo codice, sphinx cerca un catalogo di messaggi da `${package_dir}/locales/${language}/LC_MESSAGES/myextension.mo`. Per la ricerca viene utilizzato il *language*.

Added in version 1.8.

`sphinx.locale._(message: str858) → str859`

Funzione di traduzione per i messaggi sulla documentazione (menù, etichette, temi e così via). Questa funzione segue l'impostazione di *language*.

`sphinx.locale.__(message: str860) → str861`

Funzione di traduzione per i messaggi della console. Questa funzione segue le impostazioni di "locale" (*LC_ALL*, *LC_MESSAGES* e così via).

Internazionalizzazione dell'estensione (i18n) e localizzazione (l10n) utilizzando l'API i18n

Added in version 1.8.

Un'estensione può naturalmente arrivare con le traduzioni dei messaggi. Questo è brevemente riassunto nell'help `sphinx.locale.get_translation()`.

In pratica si deve:

1. Scegliere un nome per il catalogo di messaggi, che deve essere univoco. Di solito il nome dell'estensione viene utilizzato per il nome del catalogo dei messaggi.
2. Contrassegnare nei sorgenti dell'estensione tutti i messaggi come traducibili, tramite la funzione `sphinx.locale.get_translation()`, solitamente rinominata `_()`, ad esempio:

⁸⁵⁴ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁵⁵ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁵⁶ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁵⁷ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁵⁸ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁵⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁶⁰ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁶¹ <https://docs.python.org/3/library/stdtypes.html#str>

Listato 1: src/__init__.py

```
from sphinx.locale import get_translation

MESSAGE_CATALOG_NAME = 'myextension'
_ = get_translation(MESSAGE_CATALOG_NAME)

translated_text = _('Hello Sphinx!')
```

3. Impostare l'estensione per essere a conoscenza delle sue traduzioni dedicate:

Listato 2: src/__init__.py

```
def setup(app):
    package_dir = path.abspath(path.dirname(__file__))
    locale_dir = os.path.join(package_dir, 'locales')
    app.add_message_catalog(MESSAGE_CATALOG_NAME, locale_dir)
```

4. Genera il file del template del catalogo dei messaggi *.pot, solitamente nella directory dei sorgenti locale/, per esempio tramite [Babel](#)⁸⁶²:

```
$ pybabel extract --output=src/locale/myextension.pot src/
```

5. Creare cataloghi di messaggi (*.po) per ciascuna lingua di cui l'estensione fornirà la localizzazione, ad esempio tramite [Babel](#)⁸⁶³:

```
$ pybabel init --input-file=src/locale/myextension.pot --
→domain=myextension --output-dir=src/locale --locale=fr_FR
```

6. Tradurre manualmente i cataloghi dei messaggi per ciascuna lingua
7. Compila i cataloghi dei messaggi in file *.mo, per esempio tramite [Babel](#)⁸⁶⁴:

```
$ pybabel compile --directory=src/locale --domain=myextension
```

8. Assicurarsi che i file del catalogo dei messaggi vengano distribuiti quando verrà installato il pacchetto, aggiungendo una riga equivalente nell'estensione MANIFEST.in:

Listato 3: MANIFEST.in

```
recursive-include src *.pot *.po *.mo
```

Quando i messaggi sull'estensione sono stati modificati, si deve aggiornare anche il template del catalogo dei messaggi e i cataloghi dei messaggi, ad esempio tramite [Babel](#)⁸⁶⁵:

⁸⁶² <https://babel.pocoo.org/>

⁸⁶³ <https://babel.pocoo.org/>

⁸⁶⁴ <https://babel.pocoo.org/>

⁸⁶⁵ <https://babel.pocoo.org/>

```
$ pybabel extract --output=src/locale/myextension.pot src/
$ pybabel update --input-file=src/locale/myextension.pot --
  ↳ domain=myextension --output-dir=src/locale
```

Utilità

Sphinx fornisce classi di utilità e funzioni per sviluppare estensioni.

Classi base per i componenti

Queste classi base sono utili per consentire alle estensioni di ottenere facilmente componenti Sphinx (ad esempio [Config](#), [BuildEnvironment](#) e così via).

Nota: Le loro sottoclassi potrebbero non funzionare con i docutil semplici perché sono fortemente accoppiati con Sphinx.

class sphinx.transforms.**SphinxTransform**(*document*, *startnode=None*)

Una classe base delle Transform.

Rispetto a `docutils.transforms.Transform`, questa classe migliora l'accessibilità alle API Sphinx.

property app: [Sphinx](#)

Riferimento all'oggetto [Sphinx](#).

property config: [Config](#)

Riferimento all'oggetto [Config](#).

property env: [BuildEnvironment](#)

Riferimento all'oggetto [BuildEnvironment](#).

class sphinx.transforms.post_transforms.**SphinxPostTransform**(*document*,
startnode=None)

Una classe base delle post-transform.

Le post transform vengono richiamate per modificare il documento per ristrutturarlo per l'output. Risolvono i riferimenti, convertono le immagini, eseguono trasformazioni speciali per ciascun formato di output e così via. Questa classe aiuta a implementare queste post trasformazioni.

apply(***kwargs*: [Any](#)⁸⁶⁶) → [None](#)⁸⁶⁷

Sovrascrivere [override] per applicare la trasformazione all'albero del documento.

is_supported() → [bool](#)⁸⁶⁸

Controllare questa trasformazione che funziona per il builder attuale.

`run(**kwargs: Any869) → None870`

Metodo principale di post-trasformazione.

Le sottoclassi dovrebbero sovrascrivere questo metodo invece di `apply()`.

```
class sphinx.util.docutils.SphinxDirective(name, arguments, options, content,
                                           lineno, content_offset, block_text,
                                           state, state_machine)
```

Una classe base per le direttive Sphinx.

Questa classe fornisce metodi di supporto per le direttive Sphinx.

Nota: Le sottoclassi di questa classe potrebbero non funzionare con i docutil. Questa classe è fortemente accoppiata con Sphinx.

`get_location() → str871`

Preleva informazioni sulla posizione corrente per il log.

`get_source_info() → tuple872[str873, int874]`

Preleva sorgente e numero di riga

`parse_content_to_nodes(allow_section_headings: bool875 = False) → list876[Node]`

Analizza il contenuto della direttiva nei nodi.

Parametri

allow_section_headings -- Sono ammessi i titoli (sezioni) nel contenuto della direttiva? Notare che questa opzione aggira i normali controlli di Docutils sulla struttura del doctree e un uso improprio di questa opzione può portare a un doctree incoerente. In Docutils, i nodi di sezione dovrebbero essere solo figli dei nodi `Structural`, che includono i nodi `document`, `section` e `sidebar`.

Added in version 7.4.

`parse_inline(text: str877, *, lineno: int878 = -1) → tuple879[list880[Node], list881[system_message]]`

Analizza *text* come elementi in linea.

Parametri

- **text** -- Il testo da analizzare, che dovrebbe essere una riga o un paragrafo singoli. Questo non può contenere elementi strutturali (titoli, transizioni, direttive, ecc.).
- **lineno** -- Il numero di riga in cui inizia il testo interpretato.

⁸⁶⁶ <https://docs.python.org/3/library/typing.html#typing.Any>

⁸⁶⁷ <https://docs.python.org/3/library/constants.html#None>

⁸⁶⁸ <https://docs.python.org/3/library/functions.html#bool>

⁸⁶⁹ <https://docs.python.org/3/library/typing.html#typing.Any>

⁸⁷⁰ <https://docs.python.org/3/library/constants.html#None>

Ritorna

Un elenco di nodi (testo ed elementi in linea) e un elenco di `system_messages`.

Added in version 7.4.

`parse_text_to_nodes(text: str882 = "", /, *, offset: int883 = -1, allow_section_headings: bool884 = False) → list885[Node]`

Analizza *text* nei nodi.

Parametri

- **text** -- Testo, sotto forma di stringa. È accettato anche `StringList`.
- **allow_section_headings** -- Sono ammessi titoli (sezioni) in *text*? Notare che questa opzione aggira i normali controlli di Docutils sulla struttura del doctree e un uso improprio di questa opzione può portare a un doctree incoerente. In Docutils, i nodi di sezione dovrebbero essere solo figli dei nodi `Structural`, che includono i nodi `document`, `section` e `sidebar`.
- **offset** -- L'offset del contenuto.

Added in version 7.4.

`set_source_info(node: Node) → None886`

Imposta l'origine e il numero di riga sul nodo.

property config: *Config*

Riferimento all'oggetto *Config*.

property env: *BuildEnvironment*

Riferimento all'oggetto *BuildEnvironment*.

class sphinx.util.docutils.SphinxRole

Una classe base per i ruoli di Sphinx.

Questa classe fornisce metodi di supporto per i ruoli Sphinx.

⁸⁷¹ <https://docs.python.org/3/library/stdtypes.html#str>
⁸⁷² <https://docs.python.org/3/library/stdtypes.html#tuple>
⁸⁷³ <https://docs.python.org/3/library/stdtypes.html#str>
⁸⁷⁴ <https://docs.python.org/3/library/functions.html#int>
⁸⁷⁵ <https://docs.python.org/3/library/functions.html#bool>
⁸⁷⁶ <https://docs.python.org/3/library/stdtypes.html#list>
⁸⁷⁷ <https://docs.python.org/3/library/stdtypes.html#str>
⁸⁷⁸ <https://docs.python.org/3/library/functions.html#int>
⁸⁷⁹ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁸⁸⁰ <https://docs.python.org/3/library/stdtypes.html#list>
⁸⁸¹ <https://docs.python.org/3/library/stdtypes.html#list>
⁸⁸² <https://docs.python.org/3/library/stdtypes.html#str>
⁸⁸³ <https://docs.python.org/3/library/functions.html#int>
⁸⁸⁴ <https://docs.python.org/3/library/functions.html#bool>
⁸⁸⁵ <https://docs.python.org/3/library/stdtypes.html#list>
⁸⁸⁶ <https://docs.python.org/3/library/constants.html#None>

Nota: Le sottoclassi di questa classe potrebbero non funzionare con i docutil. Questa classe è fortemente accoppiata con Sphinx.

get_location() → `str`⁸⁸⁷

Preleva informazioni sulla posizione corrente per il log.

property config: `Config`

Riferimento all'oggetto `Config`.

content: `Sequence`⁸⁸⁸ [`str`⁸⁸⁹]

Un elenco di stringhe, il contenuto della direttiva per la personalizzazione (dalla direttiva "role").

property env: `BuildEnvironment`

Riferimento all'oggetto `BuildEnvironment`.

inliner: `Inliner`

L'oggetto `docutils.parsers.rst.states.Inliner`.

lineno: `int`⁸⁹⁰

Il numero di riga in cui inizia il testo interpretato.

name: `str`⁸⁹¹

Il nome del ruolo effettivamente utilizzato nel documento.

options: `dict`⁸⁹² [`str`⁸⁹³, `Any`⁸⁹⁴]

Un dizionario di opzioni direttive per la personalizzazione (dalla direttiva "role").

rawtext: `str`⁸⁹⁵

Una stringa contenente l'intero input di testo interpretato.

text: `str`⁸⁹⁶

Il contenuto del testo interpretato.

class sphinx.util.docutils.ReferenceRole

Una classe base per i ruoli dei riferimenti.

I ruoli dei riferimenti possono accettare lo stile `link title <target>` come testo per il ruolo. Il risultato analizzato; il titolo e il target del link verranno memorizzati in `self.title` e in `self.target`.

⁸⁸⁷ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁸⁸ <https://docs.python.org/3/library/collections.abc.html#collections.abc.Sequence>

⁸⁸⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁹⁰ <https://docs.python.org/3/library/functions.html#int>

⁸⁹¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁹² <https://docs.python.org/3/library/stdtypes.html#dict>

⁸⁹³ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁹⁴ <https://docs.python.org/3/library/typing.html#typing.Any>

⁸⁹⁵ <https://docs.python.org/3/library/stdtypes.html#str>

⁸⁹⁶ <https://docs.python.org/3/library/stdtypes.html#str>

disabled: `bool`⁸⁹⁷

Un valore booleano indica che il riferimento è disabilitato.

has_explicit_title: `bool`⁸⁹⁸

Un valore booleano per indicare che il ruolo ha un titolo esplicito o meno.

target: `str`⁸⁹⁹

La destinazione (target) del link per il testo interpretato.

title: `str`⁹⁰⁰

Il titolo del link per il testo interpretato.

class `sphinx.transforms.post_transforms.images.ImageConverter`(*document*,
startno-
de=None)

Una classe base per i convertitori di immagini.

Un convertitore di immagini è una specie di modulo di trasformazione di Docutils. Viene utilizzato per convertire i file immagine che non sono supportati da un builder nel formato appropriato per quel builder.

Ad esempio, *LaTeX builder* supporta PDF, PNG e JPEG come formati di immagini. Tuttavia non supporta le immagini SVG. In tal caso, l'utilizzo dei convertitori di immagini consente di includere queste immagini non supportate nel documento. Uno dei convertitori di immagini; *sphinx.ext.imgconverter* può convertire un'immagine SVG in formato PNG utilizzando Imagemagick internamente.

Ci sono tre passaggi per creare un convertitore di immagini personalizzato:

1. Creare una sottoclasse della classe `ImageConverter`
2. Sovrascrivere (override) `conversion_rules`, `is_available()` e `convert()`
3. Registrare il convertitore di immagini su Sphinx utilizzando *`Sphinx.add_post_transform()`*

convert(*_from: str*⁹⁰¹, *_to: str*⁹⁰²) → `bool`⁹⁰³

Convertire un file immagine nel formato previsto.

_from è un path del file immagine di origine e *_to* è il path del file di destinazione.

is_available() → `bool`⁹⁰⁴

Restituisce se il convertitore di immagini è disponibile o meno.

available: `bool`⁹⁰⁵ | `None`⁹⁰⁶ = `None`

Il convertitore è disponibile oppure no. Verrà compilato alla prima chiamata della build. Il risultato è condiviso nello stesso processo.

⁸⁹⁷ <https://docs.python.org/3/library/functions.html#bool>

⁸⁹⁸ <https://docs.python.org/3/library/functions.html#bool>

⁸⁹⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁹⁰⁰ <https://docs.python.org/3/library/stdtypes.html#str>

Da fare: Questo dovrebbe essere sottoposto a refactoring per non archiviare lo stato senza la variabile della classe.

conversion_rules: `list907[tuple908[str909, str910]] = []`

Una regola di conversione supportata dal convertitore di immagini. È rappresentato come un elenco di coppie di formati immagine di origine (mimetype) e di destinazione:

```
conversion_rules = [  
    ('image/svg+xml', 'image/png'),  
    ('image/gif', 'image/png'),  
    ('application/pdf', 'image/png'),  
]
```

default_priority = 200

Priorità numerica di questa trasformazione, da 0 a 999 (override).

Componenti di utilità

class sphinx.events.EventManager(app: Sphinx)

Manager di eventi per Sphinx.

add(name: str⁹¹¹) → None⁹¹²

Registra un evento Sphinx personalizzato.

connect(name: str⁹¹³, callback: Callable⁹¹⁴, priority: int⁹¹⁵) → int⁹¹⁶

Connette un gestore (handler) a un evento specifico.

disconnect(listener_id: int⁹¹⁷) → None⁹¹⁸

Disconnettere un handler.

emit(name: str⁹¹⁹, *args: Any⁹²⁰, allowed_exceptions: tuple⁹²¹[type⁹²²[Exception⁹²³], ...] = ()) → list⁹²⁴

Emette un evento Sphinx.

emit_firstresult(name: str⁹²⁵, *args: Any⁹²⁶, allowed_exceptions: tuple⁹²⁷[type⁹²⁸[Exception⁹²⁹], ...] = ()) → Any⁹³⁰

Emette un evento Sphinx e restituisce il primo risultato.

⁹⁰¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁹⁰² <https://docs.python.org/3/library/stdtypes.html#str>

⁹⁰³ <https://docs.python.org/3/library/functions.html#bool>

⁹⁰⁴ <https://docs.python.org/3/library/functions.html#bool>

⁹⁰⁵ <https://docs.python.org/3/library/functions.html#bool>

⁹⁰⁶ <https://docs.python.org/3/library/constants.html#None>

⁹⁰⁷ <https://docs.python.org/3/library/stdtypes.html#list>

⁹⁰⁸ <https://docs.python.org/3/library/stdtypes.html#tuple>

⁹⁰⁹ <https://docs.python.org/3/library/stdtypes.html#str>

⁹¹⁰ <https://docs.python.org/3/library/stdtypes.html#str>

Restituisce il risultato del primo handler che non restituisce None.

Funzioni di utilità

`sphinx.util.parsing.nested_parse_to_nodes`(*state*: *RSTState*, *text*: *str*⁹³¹ | *StringList*, *, *source*: *str*⁹³² = '<generated text>', *offset*: *int*⁹³³ = 0, *allow_section_headings*: *bool*⁹³⁴ = *True*, *keep_title_context*: *bool*⁹³⁵ = *False*) → *list*⁹³⁶[*Node*]

Analizza *text* nei nodi.

Parametri

- **state** -- Lo stato della macchina a stati. Deve essere una sottoclasse di *RSTState*.
- **text** -- Testo, sotto forma di stringa. È accettato anche *StringList*.
- **source** -- La sorgente del testo, utilizzata durante la creazione di una nuova *StringList*.
- **offset** -- L'offset del contenuto.
- **allow_section_headings** -- Sono ammessi titoli (sezioni) in *text*?
Notare che questa opzione aggira i normali controlli di Docutils sulla struttura del doctree e un uso improprio di questa opzione può portare a un doctree incoerente. In Docutils, i nodi di sezione dovrebbero essere solo figli dei nodi *Structural*, che includono i nodi *document*, *section* e *sidebar*.
- **keep_title_context** -- Se questo è *False* (il default), allora *content* viene analizzato come se fosse un documento indipendente, il che si-

⁹¹¹ <https://docs.python.org/3/library/stdtypes.html#str>
⁹¹² <https://docs.python.org/3/library/constants.html#None>
⁹¹³ <https://docs.python.org/3/library/stdtypes.html#str>
⁹¹⁴ <https://docs.python.org/3/library/typing.html#typing.Callable>
⁹¹⁵ <https://docs.python.org/3/library/functions.html#int>
⁹¹⁶ <https://docs.python.org/3/library/functions.html#int>
⁹¹⁷ <https://docs.python.org/3/library/functions.html#int>
⁹¹⁸ <https://docs.python.org/3/library/constants.html#None>
⁹¹⁹ <https://docs.python.org/3/library/stdtypes.html#str>
⁹²⁰ <https://docs.python.org/3/library/typing.html#typing.Any>
⁹²¹ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁹²² <https://docs.python.org/3/library/functions.html#type>
⁹²³ <https://docs.python.org/3/library/exceptions.html#Exception>
⁹²⁴ <https://docs.python.org/3/library/stdtypes.html#list>
⁹²⁵ <https://docs.python.org/3/library/stdtypes.html#str>
⁹²⁶ <https://docs.python.org/3/library/typing.html#typing.Any>
⁹²⁷ <https://docs.python.org/3/library/stdtypes.html#tuple>
⁹²⁸ <https://docs.python.org/3/library/functions.html#type>
⁹²⁹ <https://docs.python.org/3/library/exceptions.html#Exception>
⁹³⁰ <https://docs.python.org/3/library/typing.html#typing.Any>

gnifica che le decorazioni del titolo (ad esempio le sottolineature) non devono necessariamente corrispondere al documento circostante. Ciò è utile quando il contenuto analizzato proviene da un contesto completamente diverso, come le docstring. Se è True, e sottolineature del titolo devono corrispondere a quelle del documento circostante, altrimenti il comportamento non è definito.

Added in version 7.4.

Tipi di utility

class sphinx.util.typing.ExtensionMetadata

I metadati restituiti dalla funzione `setup()` di un'estensione.

Vedere *Metadati dell'estensione*.

env_version: `int`⁹³⁷

Un numero intero che identifica la versione dei dati env aggiunti dall'estensione.

parallel_read_safe: `bool`⁹³⁸

Indica se la lettura parallela dei file di origine è supportata dall'estensione.

parallel_write_safe: `bool`⁹³⁹

Indica se l'estensione supporta la scrittura parallela dei file di output (il default: True).

version: `str`⁹⁴⁰

La versione dell'estensione (default: 'unknown version').

API deprecate

Nello sviluppo di Sphinx, prestiamo sempre attenzione alla compatibilità delle nostre API. Ma, a volte, per alcuni motivi è necessario cambiare l'interfaccia. In questi casi, le abbiamo contrassegnate come deprecate. E vengono conservate durante le due versioni major (per maggiori dettagli, consultare *Politica sulla deprecazione*).

Di seguito è riportato un elenco delle interfacce deprecate.

⁹³¹ <https://docs.python.org/3/library/stdtypes.html#str>

⁹³² <https://docs.python.org/3/library/stdtypes.html#str>

⁹³³ <https://docs.python.org/3/library/functions.html#int>

⁹³⁴ <https://docs.python.org/3/library/functions.html#bool>

⁹³⁵ <https://docs.python.org/3/library/functions.html#bool>

⁹³⁶ <https://docs.python.org/3/library/stdtypes.html#list>

⁹³⁷ <https://docs.python.org/3/library/functions.html#int>

⁹³⁸ <https://docs.python.org/3/library/functions.html#bool>

⁹³⁹ <https://docs.python.org/3/library/functions.html#bool>

⁹⁴⁰ <https://docs.python.org/3/library/stdtypes.html#str>

Tabella 4: API deprecate

Target	De- pre- cato	Rimos- so	Alternativa
<code>sphinx.testing.util.strip_escseq</code>	7.3	9.0	<code>sphinx.util.console.strip_colors</code>
Makefile vecchio stile in <code>sphinx-quickstart</code> e le opzioni <code>-M</code> , <code>-m</code> , <code>--no-use-make-mode</code> e <code>--use-make-mode</code>	7.3	9.0	Vendor dei template dei Makefile vecchio stile ⁹⁴¹
<code>sphinx.ext.autodoc.preserve_defaults.get_function_def()</code>	7.2	9.0	N/A (la sostituzione è privata)
<code>sphinx.builders.html.StandaloneHTMLBuilder.css_files</code>	7.2	9.0	N/A
<code>sphinx.builders.html.StandaloneHTMLBuilder.script_files</code>	7.2	9.0	N/A
<code>sphinx.builders.html.Stylesheet</code>	7.2	9.0	<code>sphinx.application.Sphinx.add_css_file()</code>
<code>sphinx.builders.html.JavaScript</code>	7.2	9.0	<code>sphinx.application.Sphinx.add_js_file()</code>
<code>sphinx.util.split_into</code>	7.2	9.0	N/A
<code>sphinx.util.split_index_msg</code>	7.2	9.0	<code>sphinx.util.index_entries.split_index_msg</code>
<code>sphinx.testing.path</code>	7.2	9.0	<code>os.path</code> or <code>pathlib</code>
<code>sphinx.util.md5</code>	7.2	9.0	<code>hashlib.md5</code>
<code>sphinx.util.sha1</code>	7.2	9.0	<code>hashlib.sha1</code>
<code>sphinx.util.osutil.cd</code>	6.2	8.0	<code>contextlib.chdir</code>
<code>sphinx.util.save_traceback</code>	6.1	8.0	<code>sphinx.util.exceptions.save_traceback</code>
<code>sphinx.util.format_exception_cut_frames</code>	6.1	8.0	<code>sphinx.util.exceptions.format_exception_cut_frames</code>
<code>sphinx.util.epoch_to_rfc1123</code>	6.1	8.0	<code>sphinx.util.http_date.epoch_to_rfc1123</code>
<code>sphinx.util.rfc1123_to_epoch</code>	6.1	8.0	<code>sphinx.util.http_date.rfc1123_to_epoch</code>
<code>sphinx.util.status_iterator</code>	6.1	8.0	<code>sphinx.util.display.status_iterator</code>
<code>sphinx.util.display_chunk</code>	6.1	8.0	<code>sphinx.util.display.display_chunk</code>
<code>sphinx.util.SkipProgressMessage</code>	6.1	8.0	<code>sphinx.util.display.SkipProgressMessage</code>

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.util. progress_message	6.1	8.0	sphinx.util.display. progress_message
sphinx.util.typing. stringify	6.1	8.0	sphinx.util.typing. stringify_annotation
Supporto HTML 4	5.2	7.0	N/A
sphinx.util.path_stabilize	5.1	7.0	sphinx.util.osutil. path_stabilize
sphinx.util. get_matching_files	5.1	7.0	sphinx.util.matching. get_matching_files
sphinx.ext.napoleon. iterators	5.1	7.0	pockets.iterators
sphinx.util.stemmer	5.1	7.0	snowballstemmer
sphinx.util.jsdump	5.0	7.0	Il modulo json della libreria standard.
L'integrazione di Setuptools (setup.py build_sphinx)	5.0	7.0	N/A
L'argomento locale di sphinx.util. i18n:babel_format_date()	5.0	7.0	N/A
L'argomento language di sphinx.util. i18n:format_date()	5.0	7.0	N/A
sphinx.builders.html. html5_ready	5.0	7.0	N/A
sphinx.io.read_doc()	5.0	7.0	sphinx.builders.Builder. read_doc()
sphinx.util.docutils. __version_info__	5.0	7.0	docutils.__version_info__
sphinx.util.docutils. is_html5_writer_available()	5.0	7.0	N/A
sphinx.writers.latex. LaTeXWriter.docclasses	5.0	7.0	N/A
sphinx.ext.napoleon. docstring.GoogleDocstring. _qualify_name()	4.5	6.0	N/A
sphinx.ext.autodoc. AttributeDocumenter. _datadescriptor	4.3	6.0	N/A
sphinx.writers.html. HTMLTranslator. _fieldlist_row_index	4.3	6.0	sphinx.writers.html. HTMLTranslator. _fieldlist_row_indices

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.writers.html. HTMLTranslator. _table_row_index	4.3	6.0	sphinx.writers.html. HTMLTranslator. _table_row_indices
sphinx.writers.html5. HTML5Translator. _fieldlist_row_index	4.3	6.0	sphinx.writers.html5. HTML5Translator. _fieldlist_row_indices
sphinx.writers.html5. HTML5Translator. _table_row_index	4.3	6.0	sphinx.writers.html5. HTML5Translator. _table_row_indices
L'argomento opzionale app per sphinx.environment. BuildEnvironment	4.1	6.0	L'argomento richiesto
sphinx.application.Sphinx. html_theme	4.1	6.0	sphinx.registry. SphinxComponentRegistry. html_themes
sphinx.ext.autosummary. _app	4.1	6.0	N/A
sphinx.util.docstrings. extract_metadata()	4.1	6.0	sphinx.util.docstrings. separate_metadata()
favicon variable in HTML templates	4.0	6.0	favicon_url
Variabile logo nei template HTML	4.0	6.0	logo_url
sphinx.directives.patches. ListTable	4.0	6.0	docutils.parsers.rst. directives.tables. ListSVTable
sphinx.directives.patches. RSTTable	4.0	6.0	docutils.parsers.rst. directives.tables. RSTTable
sphinx.ext.autodoc. directive. DocumenterBridge. filename_set	4.0	6.0	sphinx.ext.autodoc. directive. DocumenterBridge. record_dependencies
sphinx.ext.autodoc. directive. DocumenterBridge.warn()	4.0	6.0	<i>API per i Log</i>
sphinx.registry. SphinxComponentRegistry. get_source_input()	4.0	6.0	N/A
sphinx.registry. SphinxComponentRegistry. source_inputs	4.0	6.0	N/A

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.transforms. FigureAligner	4.0	6.0	N/A
sphinx.util.pycompat. convert_with_2to3()	4.0	6.0	N/A
sphinx.util.pycompat. execfile_()	4.0	6.0	N/A
sphinx.util.smartypants	4.0	6.0	docutils.utils. smartquotes
sphinx.util.typing. DirectiveOption	4.0	6.0	N/A
nodo pending_xref per l'estensione viewcode	3.5	5.0	sphinx.ext.viewcode. viewcode_anchor
sphinx.builders.linkcheck. CheckExternalLinksBuilder. anchors_ignore	3.5	5.0	N/A
sphinx.builders.linkcheck. CheckExternalLinksBuilder. auth	3.5	5.0	N/A
sphinx.builders.linkcheck. CheckExternalLinksBuilder. broken	3.5	5.0	N/A
sphinx.builders.linkcheck. CheckExternalLinksBuilder. good	3.5	5.0	N/A
sphinx.builders.linkcheck. CheckExternalLinksBuilder. redirected	3.5	5.0	N/A
sphinx.builders.linkcheck. CheckExternalLinksBuilder. rqueue	3.5	5.0	N/A
sphinx.builders.linkcheck. CheckExternalLinksBuilder. to_ignore	3.5	5.0	N/A
sphinx.builders.linkcheck. CheckExternalLinksBuilder. workers	3.5	5.0	N/A
sphinx.builders.linkcheck. CheckExternalLinksBuilder. wqueue	3.5	5.0	N/A
sphinx.builders.linkcheck. node_line_or_0()	3.5	5.0	sphinx.util.nodes. get_node_line()

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.ext.autodoc. AttributeDocumenter. isinstanceattribute()	3.5	5.0	N/A
sphinx.ext.autodoc. importer. get_module_members()	3.5	5.0	sphinx.ext.autodoc. ModuleDocumenter. get_module_members()
sphinx.ext.autosummary. generate._simple_info()	3.5	5.0	<i>API per i Log</i>
sphinx.ext.autosummary. generate._simple_warn()	3.5	5.0	<i>API per i Log</i>
sphinx.writers.html. HTMLTranslator. permalink_text	3.5	5.0	<i>html_permaLinks_icon</i>
sphinx.writers.html5. HTML5Translator. permalink_text	3.5	5.0	<i>html_permaLinks_icon</i>
L'argomento follow_wrapped di sphinx.util.inspect. signature()	3.4	5.0	N/A
L'argomento no_docstring di sphinx.ext.autodoc. Documenter.add_content()	3.4	5.0	sphinx.ext.autodoc. Documenter.get_doc()
sphinx.ext.autodoc. Documenter. get_object_members()	3.4	6.0	sphinx.ext.autodoc. ClassDocumenter. get_object_members()
sphinx.ext.autodoc. DataDeclarationDocumenter	3.4	5.0	sphinx.ext.autodoc. DataDocumenter
sphinx.ext.autodoc. GenericAliasDocumenter	3.4	5.0	sphinx.ext.autodoc. DataDocumenter
sphinx.ext.autodoc. InstanceAttributeDocumenter	3.4	5.0	sphinx.ext.autodoc. AttributeDocumenter
sphinx.ext.autodoc. SlotsAttributeDocumenter	3.4	5.0	sphinx.ext.autodoc. AttributeDocumenter
sphinx.ext.autodoc. TypeVarDocumenter	3.4	5.0	sphinx.ext.autodoc. DataDocumenter
sphinx.ext.autodoc. directive. DocumenterBridge.reporter	3.5	5.0	sphinx.util.logging
sphinx.ext.autodoc. importer._getannotations()	3.4	4.0	sphinx.util.inspect. getannotations()
sphinx.ext.autodoc. importer._getmro()	3.4	4.0	sphinx.util.inspect. getmro()

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.pycode. ModuleAnalyzer.parse()	3.4	5.0	sphinx.pycode. ModuleAnalyzer.analyze()
sphinx.util.osutil. movefile()	3.4	5.0	os.replace()
sphinx.util.requests. is_ssl_error()	3.4	5.0	N/A
sphinx.builders.latex. LaTeXBuilder.usepackages	3.3	5.0	N/A
sphinx.builders.latex. LaTeXBuilder. usepackages_afger_hyperref	3.3	5.0	N/A
sphinx.ext.autodoc. SingledispatchFunctionDocume	3.3	5.0	sphinx.ext.autodoc. FunctionDocumenter
sphinx.ext.autodoc. SingledispatchMethodDocument	3.3	5.0	sphinx.ext.autodoc. MethodDocumenter
sphinx.ext.autodoc. members_set_option()	3.2	5.0	N/A
sphinx.ext.autodoc. merge_special_members_option	3.2	5.0	sphinx.ext.autodoc. merge_members_option()
sphinx.writers.texinfo. TexinfoWriter.desc	3.2	5.0	sphinx.writers.texinfo. TexinfoWriter.descs
Il primo argomento per sphinx. ext.autosummary.generate. AutosummaryRenderer stato modificato nell'oggetto Sphinx	3.1	5.0	N/A
sphinx.ext.autosummary. generate. AutosummaryRenderer accetta un tipo di oggetto come argomento	3.1	5.0	N/A
L'argomento ignore di sphinx.ext.autodoc. Documenter.get_doc()	3.1	5.0	N/A
L'argomento template_dir di sphinx.ext.autosummary. generate. AutosummaryRenderer	3.1	5.0	N/A
L'argomento module di sphinx. ext.autosummary.generate. find_autosummary_in_docstrin	3.0	5.0	N/A
L'argomento builder di sphinx. ext.autosummary.generate. generate_autosummary_docs()	3.1	5.0	N/A

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
L'argomento <code>template_dir</code> di <code>sphinx.ext.autosummary.generate.generate_autosummary_docs()</code>	3.1	5.0	N/A
<code>sphinx.ext.autosummary.generate.AutosummaryRenderer.exists()</code>	3.1	5.0	N/A
L'argomento <code>ignore</code> di <code>sphinx.util.docstring.prepare_docstring()</code>	3.1	5.0	N/A
<code>sphinx.util.rpartition()</code>	3.1	5.0	<code>str.rpartition()</code>
<code>desc_signature['first']</code>		3.0	N/A
<code>sphinx.directives.DescDirective</code>	3.0	5.0	<code>sphinx.directives.ObjectDescription</code>
<code>sphinx.domains.std.StandardDomain.add_object()</code>	3.0	5.0	<code>sphinx.domains.std.StandardDomain.note_object()</code>
<code>sphinx.domains.python.PyDecoratorMixin</code>	3.0	5.0	N/A
<code>sphinx.ext.autodoc.get_documenters()</code>	3.0	5.0	<code>sphinx.registry.documenters</code>
<code>sphinx.ext.autosummary.process_autosummary_toc()</code>	3.0	5.0	N/A
<code>sphinx.parsers.Parser.app</code>	3.0	5.0	N/A
<code>sphinx.testing.path.Path.text()</code>	3.0	5.0	<code>sphinx.testing.path.Path.read_text()</code>
<code>sphinx.testing.path.Path.bytes()</code>	3.0	5.0	<code>sphinx.testing.path.Path.read_bytes()</code>
<code>sphinx.util.inspect.getargspec()</code>	3.0	5.0	<code>inspect.getargspec()</code>
<code>sphinx.writers.latex.LaTeXWriter.format_docclass()</code>	3.0	5.0	Temi LaTeX
L'argomento <code>decode</code> di <code>sphinx.pycode.ModuleAnalyzer()</code>	2.4	4.0	N/A
<code>sphinx.directives.other.Index</code>	2.4	4.0	<code>sphinx.domains.index.IndexDirective</code>
<code>sphinx.environment.temp_data['gloss_entries']</code>	2.4	4.0	<code>documents.nameids</code>
<code>sphinx.environment.BuildEnvironment.indexentries</code>	2.4	4.0	<code>sphinx.domains.index.IndexDomain</code>

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.environment. collectors.indexentries. IndexEntriesCollector	2.4	4.0	sphinx.domains.index. IndexDomain
sphinx.io. FiletypeNotFoundError	2.4	4.0	sphinx.errors. FiletypeNotFoundError
sphinx.ext.apidoc.INITPY	2.4	4.0	N/A
sphinx.ext.apidoc. shall_skip()	2.4	4.0	sphinx.ext.apidoc. is_skipped_package
sphinx.io.get_filetype()	2.4	4.0	sphinx.util. get_filetype()
sphinx.pycode. ModuleAnalyzer.encoding	2.4	4.0	N/A
sphinx.roles.Index	2.4	4.0	sphinx.domains.index. IndexRole
sphinx.util. detect_encoding()	2.4	4.0	tokenize. detect_encoding()
sphinx.util. get_module_source()	2.4	4.0	N/A
sphinx.util.inspect. Signature	2.4	4.0	sphinx.util.inspect. signature e sphinx.util.inspect. stringify_signature()
sphinx.util.inspect. safe_getmembers()	2.4	4.0	inspect.getmembers()
sphinx.writers.latex. LaTeXTranslator.settings. author	2.4	4.0	N/A
sphinx.writers.latex. LaTeXTranslator.settings. contentsname	2.4	4.0	document['contentsname']
sphinx.writers.latex. LaTeXTranslator.settings. docclass	2.4	4.0	document['docclass']
sphinx.writers.latex. LaTeXTranslator.settings. docname	2.4	4.0	N/A
sphinx.writers.latex. LaTeXTranslator.settings. title	2.4	4.0	N/A
sphinx.writers.latex. ADDITIONAL_SETTINGS	2.4	4.0	sphinx.builders.latex. constants. ADDITIONAL_SETTINGS

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.writers.latex. DEFAULT_SETTINGS	2.4	4.0	sphinx.builders.latex. constants. DEFAULT_SETTINGS
sphinx.writers.latex. LUALATEX_DEFAULT_FONTPKG	2.4	4.0	sphinx.builders.latex. constants. LUALATEX_DEFAULT_FONTPKG
sphinx.writers.latex. PDFLATEX_DEFAULT_FONTPKG	2.4	4.0	sphinx.builders.latex. constants. PDFLATEX_DEFAULT_FONTPKG
sphinx.writers.latex. XELATEX_DEFAULT_FONTPKG	2.4	4.0	sphinx.builders.latex. constants. XELATEX_DEFAULT_FONTPKG
sphinx.writers.latex. XELATEX_GREEK_DEFAULT_FONTPKG	2.4	4.0	sphinx.builders.latex. constants. XELATEX_GREEK_DEFAULT_FONTPKG
sphinx.builders.gettext. POHEADER	2.3	4.0	sphinx/templates/gettext/ message.pot_t (file template)
sphinx.io. SphinxStandaloneReader.app	2.3	4.0	sphinx.io. SphinxStandaloneReader. setup()
sphinx.io. SphinxStandaloneReader.env	2.3	4.0	sphinx.io. SphinxStandaloneReader. setup()
sphinx.util.texescape. tex_escape_map	2.3	4.0	sphinx.util.texescape. escape()
sphinx.util.texescape. tex_hl_escape_map_new	2.3	4.0	sphinx.util.texescape. hlescape()
sphinx.writers.latex. LaTeXTranslator. no_contractions	2.3	4.0	N/A
sphinx.domains.math. MathDomain.add_equation()	2.2	4.0	sphinx.domains.math. MathDomain. note_equation()
sphinx.domains.math. MathDomain. get_next_equation_number()	2.2	4.0	sphinx.domains.math. MathDomain. note_equation()
Gli argomenti info e warn di sphinx.ext.autosummary. generate. generate_autosummary_docs()	2.2	4.0	logging.info() e logging.warning()
sphinx.ext.autosummary. generate._simple_info()	2.2	4.0	logging.info()

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.ext.autosummary. generate._simple_warn()	2.2	4.0	logging.warning()
sphinx.ext.todo. merge_info()	2.2	4.0	sphinx.ext.todo. TodoDomain
sphinx.ext.todo. process_todo_nodes()	2.2	4.0	sphinx.ext.todo. TodoDomain
sphinx.ext.todo. process_todos()	2.2	4.0	sphinx.ext.todo. TodoDomain
sphinx.ext.todo. purge_todos()	2.2	4.0	sphinx.ext.todo. TodoDomain
sphinx.builders.latex. LaTeXBuilder. apply_transforms()	2.1	4.0	N/A
sphinx.builders. _epub_base.EpubBuilder. esc()	2.1	4.0	html.escape()
sphinx.directives.Acks	2.1	4.0	sphinx.directives.other. Acks
sphinx.directives.Author	2.1	4.0	sphinx.directives.other. Author
sphinx.directives.Centered	2.1	4.0	sphinx.directives.other. Centered
sphinx.directives.Class	2.1	4.0	sphinx.directives.other. Class
sphinx.directives. CodeBlock	2.1	4.0	sphinx.directives.code. CodeBlock
sphinx.directives.Figure	2.1	4.0	sphinx.directives. patches.Figure
sphinx.directives.HList	2.1	4.0	sphinx.directives.other. HList
sphinx.directives. Highlight	2.1	4.0	sphinx.directives.code. Highlight
sphinx.directives.Include	2.1	4.0	sphinx.directives.other. Include
sphinx.directives.Index	2.1	4.0	sphinx.directives.other. Index
sphinx.directives. LiteralInclude	2.1	4.0	sphinx.directives.code. LiteralInclude
sphinx.directives.Meta	2.1	4.0	sphinx.directives. patches.Meta
sphinx.directives.Only	2.1	4.0	sphinx.directives.other. Only

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
<code>sphinx.directives.SeeAlso</code>	2.1	4.0	<code>sphinx.directives.other.SeeAlso</code>
<code>sphinx.directives.TabularColumns</code>	2.1	4.0	<code>sphinx.directives.other.TabularColumns</code>
<code>sphinx.directives.TocTree</code>	2.1	4.0	<code>sphinx.directives.other.TocTree</code>
<code>sphinx.directives.VersionChange</code>	2.1	4.0	<code>sphinx.directives.other.VersionChange</code>
<code>sphinx.domains.python.PyClassmember</code>	2.1	4.0	<code>sphinx.domains.python.PyAttribute</code> , <code>sphinx.domains.python.PyMethod</code> , <code>sphinx.domains.python.PyClassMethod</code> , <code>sphinx.domains.python.PyObject</code> e <code>sphinx.domains.python.PyStaticMethod</code>
<code>sphinx.domains.python.PyModulelevel</code>	2.1	4.0	<code>sphinx.domains.python.PyFunction</code> , <code>sphinx.domains.python.PyObject</code> e <code>sphinx.domains.python.PyVariable</code>
<code>sphinx.domains.std.StandardDomain._resolve_citation_xref()</code>	2.1	4.0	<code>sphinx.domains.citation.CitationDomain.resolve_xref()</code>
<code>sphinx.domains.std.StandardDomain.note_citations()</code>	2.1	4.0	<code>sphinx.domains.citation.CitationDomain.note_citation()</code>
<code>sphinx.domains.std.StandardDomain.note_citation_refs()</code>	2.1	4.0	<code>sphinx.domains.citation.CitationDomain.note_citation_reference()</code>
<code>sphinx.domains.std.StandardDomain.note_labels()</code>	2.1	4.0	<code>sphinx.domains.std.StandardDomain.process_doc()</code>
<code>sphinx.domains.js.JSObject.display_prefix</code>		4.3	<code>sphinx.domains.js.JSObject.get_display_prefix()</code>
<code>sphinx.environment.NoUri</code>	2.1	3.0	<code>sphinx.errors.NoUri</code>
<code>sphinx.ext.apidoc.format_directive()</code>	2.1	4.0	N/A

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.ext.apidoc. format_heading()	2.1	4.0	N/A
sphinx.ext.apidoc. make_name()	2.1	4.0	sphinx.ext.apidoc. module_join()
sphinx.ext.autodoc. importer.MockFinder	2.1	4.0	sphinx.ext.autodoc.mock. MockFinder
sphinx.ext.autodoc. importer.MockLoader	2.1	4.0	sphinx.ext.autodoc.mock. MockLoader
sphinx.ext.autodoc. importer.mock()	2.1	4.0	sphinx.ext.autodoc.mock. mock()
sphinx.ext.autosummary. autolink_role()	2.1	4.0	sphinx.ext.autosummary. AutoLink
sphinx.ext.imgmath. DOC_BODY	2.1	4.0	N/A
sphinx.ext.imgmath. DOC_BODY_PREVIEW	2.1	4.0	N/A
sphinx.ext.imgmath. DOC_HEAD	2.1	4.0	N/A
sphinx.transforms. CitationReferences	2.1	4.0	sphinx.domains.citation. CitationReferenceTransform
sphinx.transforms. SmartQuotesSkipper	2.1	4.0	sphinx.domains.citation. CitationDefinitionTransform
sphinx.util.docfields. DocFieldTransformer. preprocess_fieldtypes()	2.1	4.0	sphinx.directives. ObjectDescription. get_field_type_map()
sphinx.util.node. find_source_node()	2.1	4.0	sphinx.util.node. get_node_source()
sphinx.util.i18n. find_catalog()	2.1	4.0	sphinx.util.i18n. docname_to_domain()
sphinx.util.i18n. find_catalog_files()	2.1	4.0	sphinx.util.i18n. CatalogRepository
sphinx.util.i18n. find_catalog_source_files()	2.1	4.0	sphinx.util.i18n. CatalogRepository

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
encoding argomento di autodoc.Documenter. get_doc(), autodoc. DocstringSignatureMixin. get_doc(), autodoc. DocstringSignatureMixin. _find_signature() e autodoc.ClassDocumenter. get_doc()	2.0	4.0	N/A
arguments of EpubBuilder. build_mimetype(), EpubBuilder. build_container(), EpubBuilder. build_content(), EpubBuilder.build_toc() e EpubBuilder.build_epub()	2.0	4.0	N/A
argomento di Epub3Builder. build_navigation_doc()	2.0	4.0	N/A
L'argomento nodetype di sphinx.search. WordCollector. is_meta_keywords()	2.0	4.0	N/A
L'argomento suffix di BuildEnvironment. doc2path()	2.0	4.0	N/A
L'argomento base in stile stringa di BuildEnvironment. doc2path()	2.0	4.0	os.path.join()
sphinx.addnodes. abbreviation	2.0	4.0	docutils.nodes. abbreviation
sphinx.builders.applehelp	2.0	4.0	sphinxcontrib.applehelp
sphinx.builders.devhelp	2.0	4.0	sphinxcontrib.devhelp
sphinx.builders.epub3. Epub3Builder. validate_config_value()	2.0	4.0	sphinx.builders.epub3. validate_config_values()
sphinx.builders.html. JSONHTMLBuilder	2.0	4.0	sphinx.builders. serializinghtml. JSONHTMLBuilder

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.builders.html. PickleHTMLBuilder	2.0	4.0	sphinx.builders. serializinghtml. PickleHTMLBuilder
sphinx.builders.html. SerializingHTMLBuilder	2.0	4.0	sphinx.builders. serializinghtml. SerializingHTMLBuilder
sphinx.builders.html. SingleFileHTMLBuilder	2.0	4.0	sphinx.builders. singlehtml. SingleFileHTMLBuilder
sphinx.builders.html. WebHTMLBuilder	2.0	4.0	sphinx.builders. serializinghtml. PickleHTMLBuilder
sphinx.builders.htmlhelp	2.0	4.0	sphinxcontrib.htmlhelp
sphinx.builders.htmlhelp. HTMLHelpBuilder. open_file()	2.0	4.0	open()
sphinx.builders.qthelp	2.0	4.0	sphinxcontrib.qthelp
sphinx.cmd.quickstart. term_decode()	2.0	4.0	N/A
sphinx.cmd.quickstart. TERM_ENCODING	2.0	4.0	sys.stdin.encoding
sphinx.config. check_unicode()	2.0	4.0	N/A
sphinx.config. string_classes	2.0	4.0	[str]
sphinx.domains.cpp. DefinitionError. description	2.0	4.0	str(exc)
sphinx.domains.cpp. NoOldIdError.description	2.0	4.0	str(exc)
sphinx.domains.cpp. UnsupportedMultiCharacterCha decoded	2.0	4.0	str(exc)
sphinx.ext.autosummary. Autosummary.warn()	2.0	4.0	N/A
sphinx.ext.autosummary. Autosummary.genopt	2.0	4.0	N/A
sphinx.ext.autosummary. Autosummary.warnings	2.0	4.0	N/A
sphinx.ext.autosummary. Autosummary.result	2.0	4.0	N/A
sphinx.ext.doctest. doctest_encode()	2.0	4.0	N/A

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
<code>sphinx.ext.jsmath</code>	2.0	4.0	<code>sphinxcontrib.jsmath</code>
<code>sphinx.roles.abbr_role()</code>	2.0	4.0	<code>sphinx.roles.Abbreviation</code>
<code>sphinx.roles. emph_literal_role()</code>	2.0	4.0	<code>sphinx.roles. EmphasizedLiteral</code>
<code>sphinx.roles. menuselect_role()</code>	2.0	4.0	<code>sphinx.roles.GUILabel</code> o <code>sphinx.roles. MenuSelection</code>
<code>sphinx.roles.index_role()</code>	2.0	4.0	<code>sphinx.roles.Index</code>
<code>sphinx.roles. indexmarkup_role()</code>	2.0	4.0	<code>sphinx.roles.PEP</code> or <code>sphinx.roles.RFC</code>
<code>sphinx.testing.util. remove_unicode_literal()</code>	2.0	4.0	N/A
<code>sphinx.util.attrdict</code>	2.0	4.0	N/A
<code>sphinx.util.force_decode()</code>	2.0	5.0	N/A
<code>sphinx.util. get_matching_docs()</code>	2.0	4.0	<code>sphinx.util. get_matching_files()</code>
<code>sphinx.util.inspect. Parameter</code>	2.0	3.0	N/A
<code>sphinx.util.jsonimpl</code>	2.0	4.0	<code>sphinxcontrib. serializinghtml.jsonimpl</code>
<code>sphinx.util.osutil.EEXIST</code>	2.0	4.0	<code>errno.EEXIST</code> o <code>FileExistsError</code>
<code>sphinx.util.osutil.EINVAL</code>	2.0	4.0	<code>errno.EINVAL</code>
<code>sphinx.util.osutil.ENOENT</code>	2.0	4.0	<code>errno.ENOENT</code> o <code>FileNotFoundError</code>
<code>sphinx.util.osutil.EPIPE</code>	2.0	4.0	<code>errno.ENOENT</code> o <code>BrokenPipeError</code>
<code>sphinx.util.osutil.walk()</code>	2.0	4.0	<code>os.walk()</code>
<code>sphinx.util.pycompat. NoneType</code>	2.0	4.0	<code>sphinx.util.typing. NoneType</code>
<code>sphinx.util.pycompat. TextIOWrapper</code>	2.0	4.0	<code>io.TextIOWrapper</code>
<code>sphinx.util.pycompat. UnicodeMixin</code>	2.0	4.0	N/A
<code>sphinx.util.pycompat. html.escape()</code>	2.0	4.0	<code>html.escape()</code>
<code>sphinx.util.pycompat. indent()</code>	2.0	4.0	<code>textwrap.indent()</code>
<code>sphinx.util.pycompat. sys_encoding</code>	2.0	4.0	<code>sys.getdefaultencoding()</code>
<code>sphinx.util.pycompat. terminal_safe()</code>	2.0	4.0	<code>sphinx.util.console. terminal_safe()</code>

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
<code>sphinx.util.pycompat.u</code>	2.0	4.0	N/A
<code>sphinx.util. PeekableIterator</code>	2.0	4.0	N/A
Omissione dell'argomento <code>filename</code> in un metodo <code>IndexBuilder.feed()</code> overridden [<i>sovrascritto</i>].ù	2.0	4.0	<code>IndexBuilder. feed(docname, filename, title, doctree)</code>
<code>sphinx.writers.latex. ExtBabel</code>	2.0	4.0	<code>sphinx.builders.latex. util.ExtBabel</code>
<code>sphinx.writers.latex. LaTeXTranslator. babel_defmacro()</code>	2.0	4.0	N/A
<code>sphinx.application.Sphinx. _setting_up_extension</code>	2.0	3.0	N/A
L'argomento <code>importer</code> di <code>sphinx.ext.autodoc. importer._MockModule</code>	2.0	3.0	N/A
<code>sphinx.ext.autodoc. importer._MockImporter</code>	2.0	3.0	N/A
<code>sphinx.io. SphinxBaseFileInput</code>	2.0	3.0	N/A
<code>sphinx.io.SphinxFileInput. supported</code>	2.0	3.0	N/A
<code>sphinx.io. SphinxRSTFileInput</code>	2.0	3.0	N/A
<code>sphinx.registry. SphinxComponentRegistry. add_source_input()</code>	2.0	3.0	N/A
<code>sphinx.writers.latex. LaTeXTranslator. _make_visit_admonition()</code>	2.0	3.0	N/A
<code>sphinx.writers.latex. LaTeXTranslator. collect_footnotes()</code>	2.0	4.0	N/A
<code>sphinx.writers.texinfo. TexinfoTranslator. _make_visit_admonition()</code>	2.0	3.0	N/A
<code>sphinx.writers.text. TextTranslator. _make_depart_admonition()</code>	2.0	3.0	N/A
<code>sphinx.writers.latex. LaTeXTranslator. generate_numfig_format()</code>	2.0	4.0	N/A

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
highlightlang	1.8	4.0	<i>highlight</i>
add_stylesheet()	1.8	6.0	<i>add_css_file()</i>
add_javascript()	1.8	4.0	<i>add_js_file()</i>
<i>autodoc_default_flags</i>	1.8	4.0	<i>autodoc_default_options</i>
L'argomento content di sphinx.util.image. guess_mimetype()	1.8	3.0	N/A
L'argomento ``gettext_compact`` di sphinx.util.i18n. find_catalog_source_files()	1.8	3.0	N/A
sphinx.io. SphinxI18nReader. set_lineno_for_reporter()	1.8	3.0	N/A
sphinx.io. SphinxI18nReader.line	1.8	3.0	N/A
sphinx.directives.other. VersionChanges	1.8	3.0	sphinx.domains.changeset. VersionChanges
sphinx.highlighting. PygmentsBridge. unhighlight()	1.8	3.0	N/A
trim_doctest_flags argomento di sphinx.highlighting. PygmentsBridge	1.8	3.0	N/A
sphinx.ext.mathbase	1.8	3.0	N/A
sphinx.ext.mathbase. MathDomain	1.8	3.0	sphinx.domains.math. MathDomain
sphinx.ext.mathbase. MathDirective	1.8	3.0	sphinx.directives. patches.MathDirective
sphinx.ext.mathbase. math_role()	1.8	3.0	docutils.parsers.rst. roles.math_role()
sphinx.ext.mathbase. setup_math()	1.8	3.0	<i>add_html_math_renderer()</i>
sphinx.ext.mathbase. is_in_section_title()	1.8	3.0	N/A
sphinx.ext.mathbase. get_node_equation_number()	1.8	3.0	sphinx.util.math. get_node_equation_number()
sphinx.ext.mathbase. wrap_displaymath()	1.8	3.0	sphinx.util.math. wrap_displaymath()
sphinx.ext.mathbase.math (nodo)	1.8	3.0	docutils.nodes.math
sphinx.ext.mathbase. displaymath (nodo)	1.8	3.0	docutils.nodes.math_block

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.ext.mathbase.eqref (nodo)	1.8	3.0	sphinx.builders.latex. nodes.math_reference
viewcode_import (valore di config)	1.8	3.0	<i>viewcode_follow_imported_members</i>
sphinx.writers.latex. Table. caption_footnotetexts	1.8	3.0	N/A
sphinx.writers.latex. Table.header_footnotetexts	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. footnotestack	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. in_container_literal_block	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. next_section_ids	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. next_hyperlink_ids	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. restrict_footnote()	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. unrestrict_footnote()	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. push_hyperlink_ids()	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. pop_hyperlink_ids()	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator.bibitems	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. hlsettingstack	1.8	3.0	N/A
sphinx.writers.latex. ExtBabel. get_shorthandoff()	1.8	3.0	N/A

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.writers.html. HTMLTranslator. highlightlang()	1.8	3.0	N/A
sphinx.writers.html. HTMLTranslator. highlightlang_base()	1.8	3.0	N/A
sphinx.writers.html. HTMLTranslator. highlightlangopts()	1.8	3.0	N/A
sphinx.writers.html. HTMLTranslator. highlightlinenothreshold()	1.8	3.0	N/A
sphinx.writers.html5. HTMLTranslator. highlightlang()	1.8	3.0	N/A
sphinx.writers.html5. HTMLTranslator. highlightlang_base()	1.8	3.0	N/A
sphinx.writers.html5. HTMLTranslator. highlightlangopts()	1.8	3.0	N/A
sphinx.writers.html5. HTMLTranslator. highlightlinenothreshold()	1.8	3.0	N/A
sphinx.writers.latex. LaTeXTranslator. check_latex_elements()	1.8	3.0	Niente
sphinx.application. CONFIG_FILENAME	1.8	3.0	sphinx.config. CONFIG_FILENAME
Config.check_unicode()	1.8	3.0	sphinx.config. check_unicode()
Config.check_types()	1.8	3.0	sphinx.config. check_confval_types()
dirname, filename e tags argomenti di Config.__init__()	1.8	3.0	Config.read()
Il valore di html_search_options	1.8	3.0	vedere html_search_options
sphinx.versioning. prepare()	1.8	3.0	sphinx.versioning. UIDTransform
Sphinx.override_domain()	1.8	3.0	add_domain()
Sphinx.import_object()	1.8	3.0	sphinx.util. import_object()

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
suffix argomento di add_source_parser()	1.8	3.0	add_source_suffix()
BuildEnvironment.load()	1.8	3.0	<code>pickle.load()</code>
BuildEnvironment.loads()	1.8	3.0	<code>pickle.loads()</code>
BuildEnvironment. frompickle()	1.8	3.0	<code>pickle.load()</code>
BuildEnvironment.dump()	1.8	3.0	<code>pickle.dump()</code>
BuildEnvironment.dumps()	1.8	3.0	<code>pickle.dumps()</code>
BuildEnvironment. topickle()	1.8	3.0	<code>pickle.dump()</code>
BuildEnvironment. _nitpick_ignore	1.8	3.0	nitpick_ignore
BuildEnvironment. versionchanges	1.8	3.0	N/A
BuildEnvironment.update()	1.8	3.0	<code>Builder.read()</code>
BuildEnvironment. read_doc()	1.8	3.0	<code>Builder.read_doc()</code>
BuildEnvironment. _read_serial()	1.8	3.0	<code>Builder.read()</code>
BuildEnvironment. _read_parallel()	1.8	3.0	<code>Builder.read()</code>
BuildEnvironment. write_doctree()	1.8	3.0	<code>Builder.write_doctree()</code>
BuildEnvironment. note_versionchange()	1.8	3.0	<code>ChangesDomain. note_changeset()</code>
warn() (funzione template helper)	1.8	3.0	<code>warning()</code>
source_parsers	1.8	3.0	add_source_parser()
sphinx.util.docutils. directive_helper()	1.8	3.0	Directive classe di docutils
sphinx.cmdline	1.8	3.0	<code>sphinx.cmd.build</code>
sphinx.make_mode	1.8	3.0	<code>sphinx.cmd.make_mode</code>
sphinx.locale.l_()	1.8	3.0	sphinx.locale._()
sphinx.locale. lazy_gettext()	1.8	3.0	sphinx.locale._()
sphinx.locale.mygettext()	1.8	3.0	sphinx.locale._()
sphinx.util. copy_static_entry()	1.5	3.0	<code>sphinx.util.fileutil. copy_asset()</code>
sphinx.build_main()	1.7	2.0	<code>sphinx.cmd.build. build_main()</code>
sphinx.ext.intersphinx. debug()	1.7	2.0	<code>sphinx.ext.intersphinx. inspect_main()</code>
sphinx.ext.autodoc. format_annotation()	1.7	2.0	<code>sphinx.util.inspect. Signature</code>

continues on next page

Tabella 4 – continua dalla pagina precedente

Target	De- pre- cato	Rimos- so	Alternativa
sphinx.ext.autodoc. formatargspec()	1.7	2.0	sphinx.util.inspect. Signature
sphinx.ext.autodoc. AutodocReporter	1.7	2.0	sphinx.util.docutils. switch_source_input()
sphinx.ext.autodoc. add_documenter()	1.7	2.0	add_autodocumenter()
sphinx.ext.autodoc. AutoDirective._register	1.7	2.0	add_autodocumenter()
AutoDirective. _special_attrgetters	1.7	2.0	add_autodoc_attrgetter()
Sphinx.warn(), Sphinx.info()	1.6	2.0	<i>API per i Log</i>
BuildEnvironment. set_warnfunc()	1.6	2.0	<i>API per i Log</i>
BuildEnvironment. note_toctree()	1.6	2.0	Toctree.note() (in sphinx.environment. adapters.toctree)
BuildEnvironment. get_toc_for()	1.6	2.0	Toctree.get_toc_for() (in sphinx.environment. adapters.toctree)
BuildEnvironment. get_toctree_for()	1.6	2.0	Toctree.get_toctree_for() (in sphinx.environment. adapters.toctree)
BuildEnvironment. create_index()	1.6	2.0	IndexEntries. create_index() (in sphinx.environment. adapters.indexentries)
sphinx.websupport	1.6	2.0	sphinxcontrib-websupport ⁹⁴²
StandaloneHTMLBuilder. css_files	1.6	2.0	add_stylesheet()
document.settings. gettext_compact	1.8	1.8	gettext_compact
Sphinx.status_iterator()	1.6	1.7	sphinx.util. status_iterator()
Sphinx. old_status_iterator()	1.6	1.7	sphinx.util. old_status_iterator()
Sphinx._directive_helper()	1.6	1.7	sphinx.util.docutils. directive_helper()
sphinx.util.compat. Directive	1.6	1.7	docutils.parsers.rst. Directive
sphinx.util.compat. docutils_version	1.6	1.7	sphinx.util.docutils. __version_info__

Nota: Anche per quanto riguarda la deprecazione delle API pubbliche (funzioni e classi interne), seguiamo la policy il più possibile.

⁹⁴¹ https://github.com/sphinx-doc/sphinx/blob/v7.0.0/sphinx/templates/quickstart/Makefile_t

⁹⁴² <https://pypi.org/project/sphinxcontrib-websupport/>

Personalizzazione di LaTeX

A differenza dei *builder HTML*, il builder *latex* non beneficia dei temi preparati. La variabile *Opzioni per l'output LaTeX* e in particolare la variabile *latex_elements*, fornisce gran parte dell'interfaccia per la personalizzazione. Per esempio:

```
# inside conf.py
latex_engine = 'xelatex'
latex_elements = {
    'fontpkg': r'''
\setmainfont{DejaVu Serif}
\setsansfont{DejaVu Sans}
\setmonofont{DejaVu Sans Mono}
''',
    'preamble': r'''
\usepackage[titles]{tocloft}
\cftsetpnumwidth {1.25cm}\cftsetrmarg{1.5cm}
\setlength{\cftchapnumwidth}{0.75cm}
\setlength{\cftsecindent}{\cftchapnumwidth}
\setlength{\cftsecnumwidth}{1.25cm}
''',
    'fncychap': r'\usepackage[Bjornstrup]{fncychap}',
    'printindex': r'\footnotesize\raggedright\printindex',
}
latex_show_urls = 'footnote'
```

Nota: Tenere presente che le backslash [*barre rovesciate*] devono essere raddoppiate nelle stringhe letterali Python per evitare l'interpretazione come sequenze di escape. In alternativa, si possono utilizzare stringhe "grezze" come fatto sopra.

7.1 L'impostazione di configurazione `latex_elements`

Un dizionario che contiene frammenti LaTeX che sovrascrivono quelli Sphinx viene solitamente inserito nei file `.tex`. La sua chiave `'sphinxsetup'` è descritta *separatamente*. Permette anche configurazioni locali inserite nei file generati, tramite le direttive `raw`⁹⁴³. Per esempio, nella documentazione PDF questo capitolo ha uno stile particolare, come verrà descritto più avanti.

Tra le chiavi sovrascrivibili ci sono:

`'papersize'`

Opzione del formato carta della classe del documento (`'a4paper'` o `'letterpaper'`)

Default: `'letterpaper'`

`'pointsize'`

Opzione per la dimensione in punti della classe del documento (`'10pt'`, `'11pt'` o `'12pt'`)

Default: `'10pt'`

`'pxunit'`

Il valore di `px` uando utilizzato negli attributi dell'immagine `width` e `height`. Il valore di default è `'0.75bp'` che raggiunge `96px=1in` (in TeX `1in = 72bp = 72.27pt`). Per ottenere ad esempio `100px=1in` si usa `'0.01in'` o `'0.7227pt'` (quest'ultimo porta TeX a calcolare un valore più preciso, a causa dell'unità più piccola utilizzata nelle specifiche); per `72px=1in`, usare semplicemente `'1bp'`; per `90px=1in`, usare `'0.8bp'` o `'0.803pt'`.

Default: `'0.75bp'`

Added in version 1.5.

`'passoptionstopackages'`

Una stringa che verrà posizionata all'inizio del preambolo, progettata per contenere i comandi `\\PassOptionsToPackage{options}{foo}`.

Suggerimento: Può essere utilizzato anche per caricare i pacchetti LaTeX anticipatamente nel preambolo. Per esempio, il pacchetto `fancybox` non è compatibile con il caricamento tramite la chiave `'preamble'`, deve essere caricato prima.

Default: `''`

Added in version 1.4.

⁹⁴³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#raw>

'babel'

Inclusione del pacchetto "babel", il default è `'\\usepackage{babel}'` (la stringa della lingua del documento adatta viene passata come opzione di classe e `english` viene utilizzato se non è presente alcuna lingua). Per i documenti giapponesi, il default è una stringa vuota.

Con XeLaTeX e LuaLaTeX, Sphinx configura il documento LaTeX per usare `polyglossia`⁹⁴⁴, ma bisogna essere consapevoli che l'attuale `babel`⁹⁴⁵ ha migliorato il suo supporto per i motori Unicode negli ultimi anni e per alcune lingue potrebbe avere senso preferire `babel` a `polyglossia`.

Suggerimento: Dopo aver modificato una chiave LaTeX "core" come questa, ripulire il repertorio di build LaTeX prima della successiva build PDF, altrimenti è probabile che i file ausiliari rimasti interrompano la build.

Default: `'\\usepackage{babel}'` (`'` per i documenti in giapponese)

Cambiato nella versione 1.5: Per `latex_engine` si imposta `'xelatex'`, il default è `'\\usepackage{polyglossia}\\n\\setmainlanguage{<language>}'`.

Cambiato nella versione 1.6: `'lualatex'` utilizza le stesse impostazioni di default di `'xelatex'`

Cambiato nella versione 1.7.6: Per il francese con `xelatex` (non `lualatex`) il default è usare `babel`, non `polyglossia`. Per consentire a `lualatex` di usare `babel`, utilizzare questo:

```
latex_elements = {
    'polyglossia': '',
    'babel': r'\usepackage{babel}',
}
```

nel file `conf.py`.

'fontpkg'

Inclusione del pacchetto di font. Il default è:

```
r"""\usepackage{tgtermes}
\usepackage{tgheros}
\renewcommand\ttdefault{txxx}
"""
```

Per `'xelatex'` e `'lualatex'` tuttavia il default è quello di usare i FreeFont GNU.

Cambiato nella versione 1.2: Il default è `'` quando `language` usa lo script Cirillico.

⁹⁴⁴ <https://ctan.org/pkg/polyglossia>

⁹⁴⁵ <https://ctan.org/pkg/babel>

Cambiato nella versione 2.0: Include alcuni comandi di sostituzione dei font per supportare occasionalmente il greco o il cirillico in un documento che utilizza il motore 'pdflatex'.

Cambiato nella versione 4.0.0:

- I comandi di sostituzione dei font aggiunti nella 2.0 sono stati spostati nella chiave 'fontsubstitution', poiché la loro presenza qui rende complicato per l'utente personalizzare il valore di 'fontpkg'.
- Il default dei font è cambiato: utilizza ancora i cloni Times e Helvetica per serif e sans serif, ma tramite i font TeX migliori e più completi e i pacchetti LaTeX associati. Il font a spaziatura fissa è stato modificato per corrispondere meglio al clone del Times.

'fncychap'

Inclusione del pacchetto "fncychap" (che crea titoli di capitoli fantasiosi), il default è '`\usepackage[Bjarne]{fncychap}`' per la documentazione in inglese (questa opzione è leggermente personalizzata da Sphinx), '`\usepackage[Sonny]{fncychap}`' per documenti internazionalizzati (perché lo stile "Bjarne" utilizza numeri scritti in inglese). Altri stili "fncychap" da provare sono "Lenny", "Glenn", "Conny", "Rejne" e "Bjornstrup". Lo si può anche impostare con '' per disabilitare fncychap.

Il default è: '`\usepackage[Bjarne]{fncychap}`' per i documenti in inglese, '`\usepackage[Sonny]{fncychap}`' per i documenti internazionalizzati e '' per quelli giapponesi.

'preamble'

Contenuto aggiuntivo del preambolo. È possibile spostare tutte le macro necessarie in un file `mystyle.tex.txt` del repertorio dei sorgenti del progetto e fare in modo che LaTeX lo importi in fase di esecuzione:

```
'preamble': r'\input{mystyle.tex.txt}',
# or, if the \ProvidesPackage LaTeX macro is used in a file_
↪mystyle.sty
'preamble': r'\usepackage{mystyle}',
```

Occorre poi impostare opportunamente `latex_additional_files`, per esempio:

```
latex_additional_files = ["mystyle.sty"]
```

Non utilizzare `.tex` come suffisso, altrimenti il file verrà sottoposto al processo di creazione del PDF, utilizzare `.tex.txt` o `.sty` come negli esempi sopra.

Default: ''

'figure_align'

Allineamento delle figure "float" [*mobili*]. Ogni volta che un'immagine non rientra nella pagina corrente, verrà 'floated' [*spostata*] nella pagina successiva ma potrebbe essere preceduta da qualsiasi altro testo. Se non si vuole tale comportamento, si usa 'H' che

disabiliterà le figure mobili e posizionerà rigorosamente nell'ordine in cui appaiono nel sorgente.

Default: 'htbp' (qui, top, bottom, page)

Added in version 1.3.

'atendofbody'

Contenuto aggiuntivo del documento (subito prima degli indici).

Default: ''

Added in version 1.5.

'extrapackages'

Pacchetti LaTeX aggiuntivi. Per esempio:

```
latex_elements = {  
    'extrapackages': r'\usepackage{isodate}'  
}
```

I pacchetti LaTeX specificati verranno caricati prima del pacchetto hyperref e dei pacchetti caricati dalle estensioni Sphinx.

Suggerimento: Per caricare pacchetti LaTeX aggiuntivi dopo hyperref, si usa invece la chiave 'preamble'.

Default: ''

Added in version 2.3.

'footer'

Contenuto aggiuntivo del piè di pagina (prima degli indici).

Default: ''

Deprecato dalla versione 1.5: Usare invece la chiave 'atendofbody'.

Le chiavi che non devono essere sovrascritte se non in casi particolari sono:

'extraoptions'

Il default è la stringa vuota. Esempio: 'extraoptions': 'openany' consentirà ai capitoli (per documenti di tipo 'manual') di iniziare su qualsiasi [??tutte??] pagina.

Default: ''

Added in version 1.2.

Cambiato nella versione 1.6: Aggiunta questa documentazione.

'maxlistdepth'

LaTeX consente per default al massimo 6 livelli per elenchi annidati e ambienti simili a virgolette, con un massimo di 4 elenchi enumerati e 4 elenchi puntati. Impostando questa

chiave, ad esempio, su `'10'` (come una stringa) consentirà fino a 10 livelli nidificati (di tutti i tipi). Lasciarlo sulla stringa vuota significa obbedire al valore predefinito di LaTeX.

Avvertimento:

- L'uso di questa chiave potrebbe rivelarsi incompatibile con alcuni pacchetti LaTeX o classi di documenti speciali che personalizzano la propria lista.
- L'impostazione della chiave viene tacitamente *ignorata* se `\usepackage{enumitem}` viene eseguito all'interno del preambolo del documento. Utilizzare quindi piuttosto i comandi dedicati di questo pacchetto LaTeX.

Default: 6

Added in version 1.5.

'inputenc'

Inclusione del pacchetto "inputenc".

Default: `'\usepackage[utf8]{inputenc}'` quando si usa `pdflatex`, altrimenti `' '`.

Nota: Se si usa `utf8x` invece di `utf8` è obbligatorio estendere il preambolo LaTeX con opportuni comandi `\PreloadUnicodePage{<number>}`, come da documentazione di `utf8x` (`texdoc ucs` su un'installazione TeX basata su TeXLive). Altrimenti, nel PDF potrebbero verificarsi problemi imprevisti e forse difficili da individuare (ovvero che non causano un arresto anomalo della compilazione), in particolare per quanto riguarda i link ipertestuali.

Anche se vengono prese queste precauzioni, la creazione di PDF tramite il motore `pdflatex` potrebbe bloccarsi a causa del fatto che LaTeX upstream non è completamente compatibile con `utf8x`. Ad esempio, in determinate circostanze relative a blocchi di codice o nel tentativo di includere immagini i cui nomi di file contengono caratteri Unicode. Infatti, a partire dal 2015, LaTeX upstream con il motore `pdflatex` ha in qualche modo migliorato il supporto nativo per Unicode e sta diventando sempre più incompatibile con `utf8x`. In particolare, dalla versione LaTeX di ottobre 2019, i nomi dei file possono utilizzare caratteri Unicode e persino spazi. A livello di Sphinx ciò significa ad es. che le direttive `image`⁹⁴⁶ e `figure`⁹⁴⁷ sono ora compatibili con tali nomi di file per PDF tramite output LaTeX. Ma questo non funziona se è in uso `utf8x`.

Cambiato nella versione 1.4.3: In precedenza `'\usepackage[utf8]{inputenc}'` veniva utilizzato per tutti i compilatori.

'cmappkg'

Inclusione del pacchetto "cmap".

⁹⁴⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#image>

⁹⁴⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#figure>

Default: `'\\usepackage{cmap}'`

Added in version 1.2.

'fontenc'

Personalizzalo dal suo valore di default `'\\usepackage[T1]{fontenc}'` a:

- `'\\usepackage[X2,T1]{fontenc}'` se c'è occasionalmente bisogno di lettere cirilliche (физика частиц),
- `'\\usepackage[LGR,T1]{fontenc}'` se c'è occasionalmente bisogno di lettere greche (Σωματιδιακή φυσική).

Utilizzare piuttosto `[LGR,X2,T1]` se sono necessari entrambi.

Attenzione:

- Non utilizzare questa chiave per un *latex_engine* diverso da 'pdflatex'.
- Se la lingua principale è il greco, non utilizzare questa chiave. A partire da Sphinx 2.2.1, xelatex verrà utilizzato automaticamente come *latex_engine*.
- L'installazione di TeX potrebbe richiedere alcuni pacchetti aggiuntivi. Per esempio, su Ubuntu xenial, i pacchetti `texlive-lang-greek` e `cm-super` sono necessari affinché LGR funzioni. E per il supporto del cirillico sono necessari `texlive-lang-cyrillic` e `cm-super`.

Cambiato nella versione 1.5: Il default è `'\\usepackage{fontspec}'` quando *latex_engine* è 'xelatex'.

Cambiato nella versione 1.6: 'lualatex' usa fontspec per default come 'xelatex'.

Cambiato nella versione 2.0: 'lualatex' esegue `\defaultfontfeatures[\rmfamily,\sffamily]{}` per disabilitare le legature [ligature] TeX trasformando `<<` e `>>` come "escaping" funzionante con pdflatex/xelatex non funzionante con lualatex.

Cambiato nella versione 2.0: Rilevamento di LGR, T2A, X2 per attivare il supporto di lettere greche o cirilliche occasionali ('pdflatex').

Cambiato nella versione 2.3.0: 'xelatex' esegue `\defaultfontfeatures[\rmfamily,\sffamily]{}` per evitare le contrazioni di -- in "en-dash" [trattino] o la trasformazione delle virgolette semplici in graffe nei PDF (in paragrafi di testo non letterali) nonostante *smartquotes* sia impostato su False.

'fontsubstitution'

Ignorato se 'fontenc' non è stato configurato per usare LGR o X2 (o T2A). Nel caso in cui la chiave 'fontpkg' sia configurata per l'uso con alcuni font TeX noti per essere disponibili nelle codifiche LGR o X2, la si imposta come stringa vuota. Altrimenti si lascia il default.

Ignorato con *latex_engine* diverso da 'pdflatex'.

Added in version 4.0.0.

'textgreek'

Per il supporto di lettere greche occasionali.

Viene ignorato con 'latex', 'xelatex' o 'lualatex' come *latex_engine* e il valore di default è la stringa vuota o `'\usepackage{textalpha}'` per 'pdflatex' a seconda se la chiave 'fontenc' è stata utilizzata con LGR o meno. Solo gli utenti esperti di LaTeX potrebbero voler personalizzare questa chiave.

Può anche essere usato come `r'\usepackage{textalpha,alphabeta}'` per consentire a 'pdflatex' di supportare l'input Unicode greco nel contesto *math*. Per esempio: `:math:`\alpha`` (U+03B1) verrà visualizzato come α .

Il default è: `'\usepackage{textalpha}'` o `'` se fontenc non include l'opzione LGR option.

Added in version 2.0.

'geometry'

Inclusione del pacchetto "geometry", la definizione di default è:

```
'\usepackage{geometry}'
```

con un [dvipdfm] aggiuntivo per i documenti giapponesi. Il file di stile Sphinx LaTeX esegue:

```
\PassOptionsToPackage{hmargin=1in,vmargin=1in,marginpar=0.5in}
→{geometry}
```

che può essere personalizzato tramite le corrispondenti *opzioni 'sphinxsetup'*.

Il default è: `'\usepackage{geometry}'` (o `'\usepackage[dvipdfm]{geometry}'` per i documenti giapponesi)

Added in version 1.5.

Cambiato nella versione 1.5.2: L'opzione dvipdfm se *latex_engine* è 'latex'.

Added in version 1.5.3: Le *chiavi 'sphinxsetup' per i margini*.

Cambiato nella versione 1.5.3: La posizione nel file LaTeX è stata spostata dopo `\usepackage{sphinx}` e `\sphinxsetup{..}`, quindi anche dopo l'inserimento della chiave 'fontpkg'. Questo per gestire le opzioni di layout del foglio (paper) in un modo speciale per i documenti giapponesi: la larghezza del testo sarà impostata su un multiplo intero della larghezza di *zenkaku*, e l'altezza del testo su un multiplo intero della linea di base. Vedere la documentazione *hmargin* per ulteriori informazioni.

'hyperref'

Inclusione del pacchetto "hyperref"; carica anche il pacchetto "hypcap" e pubblica `\urlstyle{same}`. Questo viene fatto dopo che il file `sphinx.sty` è stato caricato e prima di eseguire il contenuto della chiave 'preamble'.

Attenzione: Il caricamento dei pacchetti "hyperref" e "hypcap" è obbligatorio.

Added in version 1.5: In precedenza questo veniva fatto dall'interno di `sphinx.sty`.

'maketitle'

Chiama "maketitle". Sostituire se si desidera generare una pagina del titolo con uno stile diverso.

Suggerimento: Se il valore della chiave è impostato su `r'\newcommand\sphinxbackoftitlepage{<Extra material>}\sphinxmaketitle'`, allora `<Extra material>` verrà scritto sul retro della pagina del titolo (solo 'manual' docclass).

Default: `'\sphinxmaketitle'`

Cambiato nella versione 1.8.3: L'originale `\maketitle` della classe del documento non viene sovrascritto, quindi è riutilizzabile come parte di alcune impostazioni personalizzate per questa chiave.

Added in version 1.8.3: Macro opzionale `\sphinxbackoftitlepage`. Può anche essere definita all'interno della chiave 'preamble' anziché in questa.

'releasename'

Valore che prefissa l'elemento 'release' sulla pagina del titolo. Per quanto riguarda *title* e *author* utilizzati nelle tuple di [latex_documents](#), vengono inseriti come markup LaTeX.

Default: 'Release'

'tableofcontents'

Chiama "tableofcontents". Il valore di default di `'\sphinxtableofcontents'` è un wrapper di `\tableofcontents` non modificato, personalizzabile dai pacchetti caricati dall'utente. Sovrascriverlo se si desidera generare un sommario diverso o per inserire contenuto tra la pagina del titolo e il sommario.

Default: `'\sphinxtableofcontents'`

Cambiato nella versione 1.5: In precedenza il significato stesso di `\tableofcontents` veniva modificato da Sphinx. Questo creava un'incompatibilità con pacchetti dedicati che lo modificavano anche come "tocloft" o "etoc".

'transition'

Comandi utilizzati per visualizzare le transizioni. Sovrascriverlo per visualizzare le transizioni in modo diverso.

Default: `'\n\n\bigskip\hrule\bigskip\n\n'`

Added in version 1.2.

Cambiato nella versione 1.6: Rimuove le `{}` non più necessarie dopo `\hrule`.

'makeindex'

Chiama "makeindex", l'ultima cosa prima di `\begin{document}`. Con `'\\usepackage[colums=1]{idxlayout}\\makeindex'` l'indice utilizzerà solo una colonna. Potrebbe essere necessario installare il pacchetto LaTeX `idxlayout`.

Default: `'\\makeindex'`

'printindex'

Chiama "printindex", l'ultima cosa nel file. Sovrascriverlo per generare l'indice in modo diverso, aggiungere del contenuto dopo l'indice o modificare il carattere. Poiché LaTeX utilizza la modalità a due colonne per l'indice, è spesso consigliabile impostare questa chiave su `'\\footnotesize\\raggedright\\printindex'`. Oppure, per ottenere un indice a una colonna, si usa `'\\def\\twocolumn[#1]{#1}\\printindex'` (questo trucco potrebbe fallire se si utilizza una classe di documenti personalizzata; quindi provare l'approccio `idxlayout` descritto nella documentazione della chiave 'makeindex').

Default: `'\\printindex'`

'fvset'

Personalizzazione del pacchetto LaTeX `fancyvrb`.

Il valore di default è `'\\fvset{fontsize=auto}'`, il che significa che la dimensione del carattere verrà regolata correttamente se un blocco di codice finisce in una nota a piè di pagina. Potrebbe essere necessario modificarlo se si usano font personalizzati: `'\\fvset{fontsize=\\small}'` se il carattere a spaziatura fissa è simile a Courier.

Default: `'\\fvset{fontsize=auto}'`

Added in version 1.8.

Cambiato nella versione 2.0: Per `'xelatex'` e `'lualatex'` il default è `'\\fvset{fontsize=\\small}'` poiché è adattato alle larghezze relative della famiglia FreeFont.

Cambiato nella versione 4.0.0: Cambiato il default per `'pdflatex'`. In precedenza si utilizzava `'\\fvset{fontsize=\\small}'`.

Cambiato nella versione 4.1.0: Cambiato il default per i documenti cinesi in `'\\fvset{fontsize=\\small,formatcom=\\xeCJKVerbAddon}'`

Le chiavi impostate da altre opzioni e che pertanto non devono essere sovrascritte sono:

`'docclass'` `'classoptions'` `'title'` `'release'` `'author'`

7.2 La configurazione di sphinxsetup

Added in version 1.5.

La chiave `'sphinxsetup'` di *latex_elements* fornisce un'interfaccia di personalizzazione di tipo LaTeX:

```
latex_elements = {
    'sphinxsetup': 'key1=value1, key2=value2, ...',
}
```

Il default è vuoto. Se non è vuoto, verrà passato come argomento alla macro `\sphinxsetup` all'interno del preambolo del documento, in questo modo:

```
\usepackage{sphinx}
\sphinxsetup{key1=value1, key2=value2,...}
```

I colori usati sopra sono forniti dall'opzione `svgnames` del pacchetto "xcolor":

```
latex_elements = {
    'passoptionstopackages': r'\PassOptionsToPackage{svgnames}{xcolor}
    ↪',
}
```

È possibile inserire gli utilizzi della macro LaTeX `\sphinxsetup` direttamente nel corpo del documento, tramite la direttiva `raw`. Questo capitolo ha uno stile nell'output PDF utilizzando il seguente inserimento all'inizio. Questo utilizza le chiavi descritte più avanti in *Chiavi aggiuntive 'sphinxsetup' simili a CSS*.

```
.. raw:: latex

    \begingroup
    \sphinxsetup{%
        TitleColor={named}{DarkGoldenrod},
        % pre_border-width is 5.1.0 alias for verbatimborder
        pre_border-width=2pt,
        pre_border-right-width=8pt,
        % pre_padding is a 5.1.0 alias for verbatimsep
        pre_padding=5pt,
        % Rounded boxes are new at 5.1.0
        pre_border-radius=5pt,
        % TeXcolor reminds that syntax must be as for LaTeX \definecolor
        pre_background-TeXcolor={named}{OldLace},
        % ... and since 5.3.0 also xcolor \colorlet syntax is accepted.
    ↪and we
        % can thus drop the {named}{...} thing if xcolor is.
    ↪available!
        pre_border-TeXcolor=Gold,
        % ... and even take more advantage of xcolor syntax:
        pre_border-TeXcolor=Gold!90,
        % add a shadow to code-blocks
```

(continues on next page)

(continua dalla pagina precedente)

```

pre_box-shadow=6pt 6pt,
pre_box-shadow-TeXcolor=gray!20,
%
% This 5.1.0 CSS-named option is alias for warningborder
div.warning_border-width=3pt,
% Prior to 5.1.0, padding for admonitions was not customizable
div.warning_padding=6pt,
div.warning_padding-right=18pt,
div.warning_padding-bottom=18pt,
% Assume xcolor has been loaded with its svgnames option
div.warning_border-TeXcolor=DarkCyan,
div.warning_background-TeXcolor=LightCyan,
% This one is the only option with space separated input:
div.warning_box-shadow=-12pt -12pt inset,
div.warning_box-shadow-TeXcolor=Cyan,
%
% The 5.1.0 new name would be div.attention_border-width
attentionborder=3pt,
% The 5.1.0 name here would be div.attention_border-TeXcolor
attentionBorderColor=Crimson,
% The 5.1.0 name would be div.attention_background-TeXcolor
attentionBgColor=FloralWhite,
%
% For note/hint/important/tip, the CSS syntax was added at 6.2.0
% Legacy syntax still works
noteborder=1pt,
noteBorderColor=Olive,
% But setting a background color via the new noteBgColor means
→ that
% it will be rendered using the same interface as warning type
noteBgColor=Olive!10,
% We can customize separately the four border-widths, and mimic
% the legacy "light" rendering, but now with a background color:
% div.note_border-left-width=0pt,
% div.note_border-right-width=0pt,
% Let's rather for variety use lateral borders:
div.note_border-top-width=0pt,
div.note_border-bottom-width=0pt,
%
% As long as only border width and border color are set, *and*
→ using
% for this the old interface, the rendering will be the "light"
→ one
hintBorderColor=LightCoral,
% but if we had used div.hint_border-TeXcolor or *any* CSS-named

```

(continues on next page)

(continua dalla pagina precedente)

```
% option we would have triggered the more complex "heavybox"
↪code.
}
```

E questo è posizionato alla fine del capitolo sorgente per terminare lo scope della configurazione:

```
.. raw:: latex

\endgroup
```

La sintassi LaTeX per le chiavi booleane richiede il *minuscolo* `true` o `false` ad esempio `'sphinxsetup': "verbatimwrapslines=false"`. Se si imposta la chiave booleana su `true`, `=true` è facoltativo. Gli spazi attorno alle virgole e ai segni di uguale vengono ignorati, quelli all'interno delle macro LaTeX possono essere significativi. Non utilizzare virgolette per racchiudere valori, siano essi numerici o stringhe.

bookmarksdepth

Controlla la profondità del pannello comprimibile [collapsible] dei segnalibri nel PDF. Può essere un numero (ad esempio 3) o un nome di sezione LaTeX (ad esempio `subsubsection`, cioè senza backslash). Per i dettagli, fare riferimento alla documentazione LaTeX `hyperref`.

Default: 5

Added in version 4.0.0.

hmargin, vmargin

Le dimensioni dei margini orizzontali (rispettivamente verticali), passate come opzioni `hmargin` (rispettivamente `vmargin`) al pacchetto `geometry`. Esempio:

```
'sphinxsetup': 'hmargin={2in,1.5in}, vmargin={1.5in,2in},
↪marginpar=1in',
```

I documenti giapponesi attualmente accettano solo il formato unidimensionale per questi parametri. Al pacchetto `geometry` vengono quindi passate le opzioni adatte per impostare la larghezza del testo su un multiplo esatto della larghezza di *zenkaku* e l'altezza del testo su un multiplo intero di `baselineskip`, con l'adattamento più vicino ai margini.

Default: 1in (equivalente a `{1in,1in}`)

Suggerimento: Per la docclass giapponese `'manual'` con `pointsize 11pt` or `12pt`, utilizzare l'opzione extra document class `nomag` (cfr. la chiave `'extraclassoptions'` di `latex_elements`) o le cosiddette unità `"true"` di TeX:

```
'sphinxsetup': 'hmargin=1.5truein, vmargin=1.5truein, marginpar=5zw
↪',
```

Added in version 1.5.3.

marginpar

La dimensione LaTeX `\marginparwidth`. Per i documenti giapponesi, il valore viene modificato per essere il multiplo intero più vicino della larghezza *zenkaku*.

Default: `0.5in`

Added in version 1.5.3.

verbatimwithframe

Booleano per specificare se i *code-block* e le include letterali sono racchiusi in frame. Impostandolo su `false` non si disattiva l'uso del pacchetto "framed", perché è ancora in uso per il colore di sfondo opzionale.

Default: `true`.

verbatimwraplines

Booleano per specificare se le righe lunghe nei contenuti di *code-block* vengono spostate a capo.

Se `true`, le interruzioni di riga possono verificarsi negli spazi (l'ultimo spazio prima dell'interruzione di riga verrà reso utilizzando un simbolo speciale) e nei caratteri di punteggiatura ASCII (cioè non nelle lettere o nelle cifre). Ogni volta che una stringa lunga non ha punti di interruzione, viene spostata alla riga successiva. Se la sua lunghezza è maggiore della larghezza della linea, traboccherà (overflow).

Default: `true`

verbatimforcewraps

Booleano per specificare se le righe lunghe nel contenuto dei *code-block* devono essere forzatamente mandate a capo per non andare in overflow.

Nota: Si presuppone che [Pygments](https://pygments.org/)⁹⁴⁸ LaTeXFormatter non sia stato utilizzato con i suoi `texcomments` o opzioni simili che consentono markup LaTeX aggiuntivo (arbitrario).

Inoltre, nel caso di *latex_engine* impostato su `'pdflatex'`, è consentita solo la gestione LaTeX di default dei punti di codice Unicode, ovvero `utf8` non `utf8x`.

Default: `false`

Added in version 3.5.0.

verbatimmaxoverfull

Un numero. Se una stringa lunga e non divisibile ha una lunghezza maggiore della larghezza totale della riga più questo numero di caratteri, e se è attiva la modalità `verbatimforcewraps`, a riga di input verrà resettata utilizzando l'algoritmo forzoso che applica punti di interruzione a ciascun carattere.

⁹⁴⁸ <https://pygments.org/>

Default: 3

Added in version 3.5.0.

verbatimmaxunderfull

Un numero. Se si applica la modalità `verbatimforcewraps` e se, dopo aver applicato il ritorno a capo degli spazi e della punteggiatura, nella prima parte della riga divisa manca almeno quel numero di caratteri per riempire la larghezza disponibile, la riga di input verrà reimpostata utilizzando l'algoritmo forzoso.

Poiché il valore di default è impostato su un valore elevato, l'algoritmo forzoso viene attivato solo in caso di overfull, ovvero in presenza di una stringa più lunga dell'intera larghezza della riga. Impostarlo su 0 per forzare il ritorno a capo di tutte le righe di input alla larghezza di riga attualmente disponibile:

```
latex_elements = {
    'sphinxsetup': "verbatimforcewraps, verbatimmaxunderfull=0",
}
```

Questo può essere fatto localmente per un dato blocco di codice tramite l'uso di direttive latex pure per inserire `\sphinxsetup` adatto (prima e dopo) nel file latex.

Default: 100

Added in version 3.5.0.

verbatimhintsturnover

Booleano per specificare se i code-block visualizzano i suggerimenti "continua nella pagina successiva" e "continua dalla pagina precedente" in caso di interruzioni di pagina.

Default: true

Added in version 1.6.3.

Cambiato nella versione 1.7: il default è cambiato da false a true.

verbatimcontinuedalign, verbatimcontinuesalign

Posizione orizzontale relativa al contenuto della cornice (frame): l (allineato a sinistra), r (allineato a destra) o c (centrato).

Default: r

Added in version 1.7.

parsedliteralwraps

Booleano per specificare se le righe lunghe nel contenuto dei `parsed-literal`⁹⁴⁹ devono andare a capo.

Default: true

Added in version 1.5.2: impostare questo valore di opzione su false per ripristinare il comportamento precedente.

⁹⁴⁹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#parsed-literal>

inlineliteralwraps

Booleano per specificare se le interruzioni di riga sono consentite all'interno dei letterali inline: ma potenziali punti di interruzione aggiuntivi (oltre a quelli consentiti da LaTeX negli spazi o per la sillabazione) sono attualmente inseriti solo dopo i caratteri . , ; ? ! / e \. A causa di fatti interni di TeX, lo spazio bianco nella riga verrà allungato (o ridotto) per accogliere l'interruzione di riga.

Default: `true`

Added in version 1.5: impostare questo valore di opzione su `false` per ripristinare il comportamento precedente.

Cambiato nella versione 2.3.0: aggiunto un potenziale punto di interruzione ai caratteri \.

verbatimvisiblepace

Quando una lunga riga di codice viene divisa, l'ultimo carattere di spazio dalla riga di codice sorgente subito prima della posizione di interruzione di riga viene composto utilizzando questo.

Default: `\textcolor{red}{\textvisiblespace}`

verbatimcontinued

Una macro LaTeX inserita all'inizio delle righe di codice di continuazione. La sua (complicata...) impostazione di default presenta un piccolo gancio rosso che punta a destra:

```
\makebox[2\fontcharwd\font`x][r]{\textcolor{red}{\tiny$↪hookrightarrow$}}
```

Cambiato nella versione 1.5: L'interruzione delle lunghe righe di codice è stata aggiunta alla versione 1.4.2. La definizione di default del simbolo di continuazione è stata modificata nella 1.5 per adattarsi a varie dimensioni di carattere (ad esempio, i blocchi di codice possono essere nelle note a piè di pagina).

Nota: I valori per le chiavi del colore devono:

- obbedire alla sintassi del comando LaTeX `\definecolor`, ad es. qualcosa come `VerbatimColor={rgb}{0.2,0.3,0.5}` o `{RGB}{37,23,255}` o `{gray}{0.75}` o (solo con il pacchetto `xcolor`) `{HTML}{808080}` o ...
- oppure obbedire alla sintassi del comando `\colorlet` dal pacchetto `xcolor` (che poi deve esistere nell'installazione di LaTeX), ad es. `VerbatimColor=red!10` o `red!50!green` o `-red!75` o `MyPreviouslyDefinedColor` o... Fare riferimento alla documentazione [xcolor](https://ctan.org/pkg/xcolor)⁹⁵⁰ per questa sintassi.

Cambiato nella versione 5.3.0: Precedentemente veniva accettata solo la sintassi di `\definecolor`.

⁹⁵⁰ <https://ctan.org/pkg/xcolor>

TitleColor

Il colore per i titoli (come configurato tramite l'uso del pacchetto "titlesec").

Default: `{rgb}{0.126,0.263,0.361}`

InnerLinkColor

Un colore passato a `hyperref` come valore di `linkcolor` e di `citecolor`.

Default: `{rgb}{0.208,0.374,0.486}`.

OuterLinkColor

Un colore passato a `hyperref` come valore di `filecolor`, `menucolor` e di `urlcolor`.

Default: `{rgb}{0.216,0.439,0.388}`

VerbatimColor

Il colore di background (sfondo) per i *code-block*.

Default: `{gray}{0.95}` (grigio)

Cambiato nella versione 6.0.0: Precedentemente era `{rgb}{1,1,1}` (bianco).

VerbatimBorderColor

Il colore della cornice (frame).

Default: `{RGB}{32,32,32}`

Cambiato nella versione 6.0.0: Precedentemente era `{rgb}{0,0,0}` (nero).

VerbatimHighlightColor

Il colore per le righe evidenziate.

Default: `{rgb}{0.878,1,1}`

Added in version 1.6.6.

TableRowColorHeader

Imposta il colore di sfondo (background) per (tutte) le righe di intestazione delle tabelle.

Avrà effetto solo se *latex_table_style* contiene 'colorrows' o se alla tabella è assegnata la classe `colorrows`. Viene ignorato per le tabelle con classe `nocolorrows`.

Come per le altre chiavi 'sphinxsetup', anche questa può essere impostata o modificata da un comando LaTeX `\sphinxsetup{...}` inserito tramite la direttiva *raw*⁹⁵¹ directive, o anche da un Ambiente LaTeX associato ad una *classe container* e che utilizza tale `\sphinxsetup{...}`.

Default: `{gray}{0.86}`

C'è anche `TableMergeColorHeader`. Se utilizzato, imposta un colore specifico per le celle unite nell'intestazione.

Added in version 5.3.0.

TableRowColorOdd

Imposta il colore di background per le righe dispari nelle tabelle (il conteggio delle righe

⁹⁵¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#raw>

inizia da 1 nella prima riga non di intestazione). Ha effetto solo se `latex_table_style` contiene 'colorrows' o per tabelle specifiche a cui è assegnata la classe `colorrows`.

Default: `{gray}{0.92}`

C'è anche `TableMergeColorOdd`.

Added in version 5.3.0.

TableRowColorEven

Imposta il colore background per le righe pari nelle tabelle.

Default `{gray}{0.98}`

C'è anche `TableMergeColorEven`.

Added in version 5.3.0.

verbatimsep

La separazione tra le righe di codice e il frame (cornice).

Vedere *Chiavi aggiuntive 'sphinxsetup' simili a CSS* per il suo alias `pre_padding` e la altre chiavi.

Default: `\fboxsep`

verbatimborder

La larghezza del frame attorno ai *code-block*. Vedere anche *Chiavi aggiuntive 'sphinxsetup' simili a CSS* per `pre_border-width`.

Default: `\fboxrule`

shadowsep

La separazione tra contenuto e frame per i riquadri (bx) *contenuti*⁹⁵² e *topic*⁹⁵³.

Vedere *Chiavi aggiuntive 'sphinxsetup' simili a CSS* per l'alias `div.topic_padding`.

Default: 5pt

shadowsize

La larghezza dell'"ombra" laterale a destra e in basso.

Vedere *Chiavi aggiuntive 'sphinxsetup' simili a CSS* per `div.topic_box-shadow` che permette di configurare separatamente la larghezza delle ombre verticali e orizzontali.

Default: 4pt

Cambiato nella versione 6.1.2: Corretta una regressione introdotta nella 5.1.0 che modificava involontariamente la larghezza dei riquadri degli argomenti e, peggio ancora, faceva uso di questa chiave interrompendo le build PDF.

shadowrule

La larghezza della cornice (frame) attorno ai riquadri (box) *topic*⁹⁵⁴. Vedere anche *Chiavi aggiuntive 'sphinxsetup' simili a CSS* per `div.topic_border-width`.

⁹⁵² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

⁹⁵³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

⁹⁵⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

Default: `\fboxrule`

noteBorderColor, hintBorderColor, importantBorderColor, tipBorderColor

Il colore per le due righe orizzontali usate da Sphinx in LaTeX per stilizzare un'ammonizione di tipo `note`⁹⁵⁵.

Default: `{rgb}{0,0,0}` (black)

noteBgColor, hintBgColor, importantBgColor, tipBgColor

Il colore opzionale per il background. È impostato a priori su bianco, ma non viene utilizzato, a meno che non sia stato impostato esplicitamente, e ciò fa sì che Sphinx passi al codice LaTeX più complesso che viene utilizzato anche per avvisi di tipo `warning`. Ci sono poi opzioni aggiuntive descritte in *Chiavi aggiuntive 'sphinxsetup' simili a CSS*.

Default: `{rgb}{1,1,1}` (white)

Added in version 6.2.0.

noteTextColor, hintTextColor, importantTextColor, tipTextColor

Il colore facoltativo per i contenuti.

Default: non impostato (utilizza il colore del testo ambientale, a priori il nero)

Added in version 6.2.0: Da considerarsi sperimentale fino alla 7.0.0. Queste opzioni hanno gli alias `div.note_TeXcolor` (ecc.) descritte in *Chiavi aggiuntive 'sphinxsetup' simili a CSS*. L'utilizzo di quest'ultimo consentirà a Sphinx di passare a un codice LaTeX più complesso, che supporta la personalizzazione descritta in *Chiavi aggiuntive 'sphinxsetup' simili a CSS*.

noteTeXextras, hintTeXextras, importantTeXextras, tipTeXextras

Del codice LaTeX extra (come `\bfseries` o `\footnotesize`) da eseguire all'inizio del contenuto.

Default: empty

Added in version 6.2.0: Da considerarsi sperimentale fino alla 7.0.0. Queste opzioni hanno gli alias `div.note_TeXextras` (ecc.) descritti in *Chiavi aggiuntive 'sphinxsetup' simili a CSS*.

noteborder, hintborder, importantborder, tipborder

La larghezza delle due righe orizzontali.

Se è impostato il colore di background, o viene utilizzata la sintassi alternativa *Chiavi aggiuntive 'sphinxsetup' simili a CSS* (ad esempio `div.note_border-width=1pt` al posto di `noteborder=1pt`), o *qualsiasi* opzione con un nome simile a CSS, allora il bordo è un frame intero e questo parametro imposta la sua larghezza anche per sinistra e destra.

Default: `0.5pt`

warningBorderColor, and (caution|attention|danger|error)BorderColor

Il colore della cornice dell'ammonizione.

Default: `{rgb}{0,0,0}` (black)

⁹⁵⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#note>

warningBgColor, cautionBgColor, attentionBgColor, dangerBgColor, errorBgColor

I colori di sfondo per le rispettive ammonizioni.

Default: `{rgb}{1,1,1}` (white)

warningborder, cautionborder, attentionborder, dangerborder, errorborder

La larghezza del frame. Vedere *Chiavi aggiuntive 'sphinxsetup' simili a CSS* per le chiavi che permettono di configurare separatamente la larghezza di ciascun bordo.

Default: 1pt

AtStartFootnote

Macro LaTeX inserite all'inizio del testo della nota a piè di pagina, dopo il numero della nota.

Default: `\mbox{ }`

BeforeFootnote

Macro LaTeX inserite prima del segno della nota a piè di pagina. Il default rimuove l'eventuale spazio prima di esso (altrimenti, TeX vi potrebbe inserire un'interruzione di riga).

Default: `\leavevmode\unskip`

Added in version 1.5.

HeaderFamily

default `\sffamily\bfseries`. Imposta il font usato nei titoli.

7.3 Chiavi aggiuntive 'sphinxsetup' simili a CSS

Added in version 5.1.0: Per *code-block*, *topic*⁹⁵⁶ e direttiva *contents*⁹⁵⁷ e ammonizioni "forti" (*warning*⁹⁵⁸, *error*⁹⁵⁹, ...).

Added in version 6.2.0: Anche per le ammonizioni *note*⁹⁶⁰, *hint*⁹⁶¹, *important*⁹⁶² e *tip*⁹⁶³ si possono definire stili in questo modo. L'utilizzo di *qualsiasi* delle opzioni elencate attiverà l'utilizzo di un codice LaTeX più complesso di quello utilizzato per default (*sphinxheavybox* rispetto a *sphinxlightbox*). L'impostazione del nuovo *noteBgColor* (o *hintBgColor*, ...) attiva anche l'utilizzo di *sphinxheavybox* per *note*⁹⁶⁴ (o *hint*⁹⁶⁵, ...).

Forse in futuro queste nuove impostazioni della 5.1.0 (e 6.2.0) verranno facoltativamente importate da qualche file CSS esterno autentico, ma attualmente devono essere utilizzate tramite

⁹⁵⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

⁹⁵⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

⁹⁵⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#warning>

⁹⁵⁹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#error>

⁹⁶⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#note>

⁹⁶¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#hint>

⁹⁶² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#important>

⁹⁶³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#tip>

⁹⁶⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#note>

⁹⁶⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#hint>

l'interfaccia 'sphinxsetup' (o il comando LaTeX `\sphinxsetup` inserito tramite la direttiva `raw`⁹⁶⁶) e la sintassi CSS viene solo imitata.

Importante: Possono verificarsi errori LaTeX di basso livello che causano un errore di compilazione se la sintassi di input non viene rispettata.

- In particolare i colori devono essere inseriti come per le altre opzioni relative ai colori precedentemente descritte, ovvero nella sintassi `\definecolor` o, se è disponibile il pacchetto `xcolor` (viene quindi utilizzato automaticamente) anche nella sintassi `\colorlet`:

```
...<other options>
div.warning_border-TeXcolor={rgb}{1,0,0},% (always works)
div.error_background-TeXcolor=red!10,% (works only if xcolor is
↪available)
...<other options>
```

- I due punti al posto del segno uguale interromperanno `[break]` LaTeX.
- `...border-width` e `...padding` prevedono una dimensione *singola*: non possono essere utilizzati finora con dimensioni separate da spazi.
- `...top-right-radius` e altri valori possono essere dimensioni separate da spazi singoli o da *due*.
- Le specifiche delle dimensioni devono utilizzare unità TeX come `pt` o `cm` o `in`. L'unità `px` è riconosciuta da `pdflatex` e da `lualatex` ma non da `xelatex` né da `platex`.
- È consentito che tali indicazioni siano cosiddette "espressioni dimensionali", ad es. `\fboxsep+2pt` e `0.5\baselineskip` sono input validi. Le espressioni verranno valutate solo al momento della stesura [typesetting time]. Fare attenzione però se si usano come in questi esempi sequenze di controllo TeX per raddoppiare la barra rovesciata o per impiegare una stringa Python pura per il valore della chiave '*sphinxsetup*'.
- Di norma, evitare di inserire spazi non necessari nei valori delle chiavi: soprattutto per i raggi un input come `2 pt 3pt` interromperà LaTeX. Attenzione inoltre che `\fboxsep` `\fboxsep` non verrà visto come separato da spazi in LaTeX. Si deve usare qualcosa come `{\fboxsep} \fboxsep`. Oppure utilizzare direttamente `3pt 3pt` che è equivalente a priori e più semplice.

Le opzioni sono tutte denominate in uno schema simile che dipende da un prefisso, seguito poi da un carattere di sottolineatura, poi dal nome della proprietà.

⁹⁶⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#raw>

Direttiva	Prefisso dell'opzione	Ambiente LaTeX
<code>code-block</code>	<code>pre</code>	<code>sphinxVerbatim</code>
<code>topic</code> ⁹⁶⁷	<code>div.topic</code>	<code>sphinxShadowBox</code>
<code>contenuti</code> ⁹⁶⁸	<code>div.topic</code>	<code>sphinxShadowBox</code>
<code>note</code> ⁹⁶⁹	<code>div.note</code>	<code>sphinxnote</code> using <code>sphinxheavybox</code>
<code>warning</code> ⁹⁷⁰	<code>div.warning</code>	<code>sphinxwarning</code> (uses <code>sphinxheavybox</code>)
tipo (type) di ammonizione	<code>div.<type></code>	<code>sphinx<type></code> (using <code>sphinxheavybox</code>)

Ecco ora queste opzioni e i loro default comuni. Sostituire di seguito `<prefix>` con il prefisso effettivo come spiegato sopra. Non dimenticare il carattere di sottolineatura che separa il prefisso dai nomi delle proprietà.

- `<prefix>_border-top-width`,
`<prefix>_border-right-width`,
`<prefix>_border-bottom-width`,
`<prefix>_border-left-width`,
`<prefix>_border-width`. Quest'ultima può (attualmente) essere solo una *singola* dimensione che quindi definisce tutte le altre quattro.

Il default è che tutte queste dimensioni siano uguali. Sono impostati su:

- `\fboxrule` (cioè a priori 0.4pt) per `code-block`,
- `\fboxrule` per `topic`⁹⁷¹ o direttiva `contents`⁹⁷²,
- 1pt per `warning`⁹⁷³ e altre ammonizioni "forti",
- 0.5pt per `note`⁹⁷⁴ e altre ammonizioni "lievi". Lo stile dei frame delle "lightbox" utilizzato per loro in assenza dell'uso delle opzioni con nome CSS verrà emulato dal più ricco "heavybox" se si imposta `border-left-width` e `border-right-width` entrambi a 0pt.
- `<prefix>_box-decoration-break` può essere impostato su `clone` o su `slice` e configura il comportamento alle interruzioni di pagina. Il default è `slice` per `code-block` (cioè per `<prefix>=pre`) dalla versione 6.0.0. Per le altre direttive il default è `clone`.
- `<prefix>_padding-top`,
`<prefix>_padding-right`,
`<prefix>_padding-bottom`,

⁹⁶⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

⁹⁶⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

⁹⁶⁹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#note>

⁹⁷⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#warning>

⁹⁷¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

⁹⁷² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

⁹⁷³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#warning>

⁹⁷⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#note>

`<prefix>_padding-left`,

`<prefix>_padding`. Quest'ultima può (attualmente) essere solo una *singola* dimensione che quindi definisce tutte le altre quattro.

Il default è che tutte queste dimensioni siano uguali. Sono impostati su:

- `\fboxsep` (cioè a priori 3pt) per *code-block*,
- 5pt per *topic*⁹⁷⁵ o per direttiva *contents*⁹⁷⁶,
- un valore speciale per *warning*⁹⁷⁷ e altre ammonizioni "forti", che garantisce un comportamento compatibile con le versioni precedenti.

Importante: Prima della versione 5.1.0 non esisteva una personalizzazione separata del riempimento (padding) per le caselle di tipo warning in PDF tramite output LaTeX. La somma del padding (*riempimento*) e della larghezza del bordo (come impostata ad esempio per *warning*⁹⁷⁸ da `warningborder`, ora chiamato anche `div.warning_border-width`) è stata mantenuta a un certo valore costante. Ciò limitava la larghezza del bordo a valori piccoli, altrimenti il bordo poteva sovrapporsi al contenuto del testo. Questo comportamento viene mantenuto come default.

- lo stesso comportamento del padding [*riempimento*] viene rispettato per default per *note*⁹⁷⁹ e altre ammonizioni "lievi" quando si usa *sphinxheavybox*.
- `<prefix>_border-top-left-radius`,
- `<prefix>_border-top-right-radius`,
- `<prefix>_border-bottom-right-radius`,
- `<prefix>_border-bottom-left-radius`,
- `<prefix>_border-radius`. Quest'ultima chiave imposta i primi quattro sul valore assegnato. Ciascun valore chiave può essere una o *due* dimensioni, separate da spazi.

Il default è che tutti e quattro gli angoli siano circolari o dritti, con raggi comuni:

- `\fboxsep` (cioè a priori 3pt) per *code-block* (dalla 6.0.0).
- 0pt per tutte le altre direttive; questo significa usare angoli dritti.

Vedere un'osservazione sopra sulle trappole con gli spazi in LaTeX.

- `<prefix>_box-shadow` è speciale nella misura in cui può essere:
 - la parola chiave *none*,
 - o una singola dimensione (che fornisce sia l'offset x che quello y),
 - o due dimensioni (separate da uno spazio),
 - o due dimensioni seguite dalla parola chiave *inset*.

⁹⁷⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

⁹⁷⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

⁹⁷⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#warning>

⁹⁷⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#warning>

⁹⁷⁹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#note>

L'offset x e quello y potrebbero essere negativi. Il default è *none*, *except* per il `topic`⁹⁸⁰ o per le direttive `contents`⁹⁸¹, per le quali è `4pt 4pt`, cioè l'ombra ha una larghezza di `4pt` e si estende fino a destra e sotto la cornice. L'ombra laterale si estende quindi nel margine destro della pagina.

- `<prefix>_border-TeXcolor`,
`<prefix>_background-TeXcolor`,
`<prefix>_box-shadow-TeXcolor`,
`<prefix>_TeXcolor`. Questi sono i colori.

Il colore di default dell'ombra in tutti i casi è `{rgb}{0,0,0}` cioè nero.

Dalla versione 6.0.0 il colore del bordo e il colore di background del `code-block`, cioè il prefisso `pre`, sono predefiniti rispettivamente su `{RGB}{32,32,32}` e `{gray}{0.95}`. In precedenza erano impostati rispettivamente sul nero e sul bianco.

Per tutti gli altri tipi, il colore del bordo predefinito è il nero e il colore dello sfondo il bianco.

Il `<prefix>_TeXcolor` sta per la proprietà CSS "color", cioè influenza il colore del testo dei contenuti. Per quanto riguarda le altre tre opzioni, il nome `TeXcolor` serve a sottolineare che la sintassi di input è quella di TeX per i colori e non quella di HTML/CSS. Se il pacchetto `xcolor` è disponibile nell'installazione LaTeX, è possibile utilizzare direttamente i colori con nome come valori chiave. Considerare il passaggio di opzioni come `dvipsnames`, `svgnames` e `x11names` a `xcolor` tramite la chiave 'passoptionstopackages' di `latex_elements`.

Se è impostato `<prefix>_TeXcolor`, viene inserito un comando `\color` all'inizio del contenuto della direttiva; per le ammonizioni avviene dopo l'intestazione che riproduce la tipologia dell'ammonizione.

- `<prefix>_TeXextras`: se impostato, il suo valore deve essere uno o più comandi LaTeX, ad esempio `\itshape`. Questi comandi verranno inseriti all'inizio del contenuto; per le ammonizioni avviene dopo l'intestazione che riproduce la tipologia dell'ammonizione.

Nota:

- Tutte le direttive supportano l'impostazione di `box-decoration-break` su `slice`.
 Cambiato nella versione 6.2.0: Precedentemente lo faceva solo `code-block`. L'impostazione di default resta `clone` per tutte le altre direttive, ma probabilmente cambierà a partire dalla versione 7.0.0.
- Gli angoli dei box arrotondati possono essere ellittici.
 Cambiato nella versione 6.2.0: In precedenza, erano supportati solo gli angoli circolari arrotondati e un angolo arrotondato obbligava l'intero frame a utilizzare la stessa larghezza costante da `<prefix>_border-width`.

⁹⁸⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

⁹⁸¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

- Le ombre inserite (`inset`) sono incompatibili con gli angoli arrotondati. Nel caso in cui siano specificati entrambi, l'ombra inserita (`inset`) verrà semplicemente ignorata.

Cambiato nella versione 6.2.0: Precedentemente, al contrario, gli angoli arrotondati venivano ignorati nel caso in cui fosse specificata un'ombra `inset`.

- `<prefix>_TeXcolor` e `<prefix>_TeXextras` sono nuovi con la versione 6.2.0.

L'utilità è dubbia nel caso di `code-block`:

- `pre_TeXcolor` influenzerà solo i pochi token evidenziati non-Pygments; colorare i numeri di riga, ma se si vuole colorarli *solo* bisogna passare attraverso l'interfaccia `fancyvrb`.
- `pre_TeXextras=\footnotesize` ad esempio può essere sostituito con l'uso utilizzando la chiave dei `latex_elements`: `'fvset'`. Per `'lualatex'` e `'xelatex'` Sphinx include nel preambolo già `\fvset{fontsize=\small}` e questo induce `fancyvrb` a sovrascrivere un `\footnotesize` proveniente da `pre_TeXextras`. Bisogna usare la sintassi `pre_TeXextras=\fvset{fontsize=\footnotesize}`. È più semplice impostare direttamente la chiave dei `latex_elements`, `'fvset'`...

Considerare queste opzioni sperimentali e che alcuni dettagli di implementazione potrebbero cambiare. Ad esempio, se i comandi LaTeX `pre_TeXextras` fossero inseriti da Sphinx in un'altra posizione, potrebbero sovrascrivere l'effetto `'fvset'`, forse questo è ciò che verrà fatto in una versione futura.

- I riquadri arrotondati vengono eseguiti utilizzando l'interfaccia `pict2e`⁹⁸² per alcune operazioni grafiche PDF di base. Se questo pacchetto LaTeX non può essere trovato, la creazione procederà e renderà tutti i riquadri con angoli dritti.
- Gli angoli ellittici utilizzano il pacchetto LaTeX `ellipse`⁹⁸³ che estende `pict2e`⁹⁸⁴. Se questo pacchetto LaTeX non può essere trovato, gli angoli arrotondati saranno archi circolari (o dritti se `pict2e`⁹⁸⁵ non è disponibile).

Il seguente comportamento legacy non è attualmente personalizzabile:

- Per `code-block`, il padding, la larghezza del bordo e l'ombra (se se ne aggiunge una) andranno nel margine; le righe di codice rimangono nella stessa posizione indipendentemente dai valori di padding e larghezza del bordo, tranne che per essere spostate verticalmente ovviamente per non sovrascrivere altro testo a causa della larghezza del bordo o dell'ombra esterna.
- Per `topic`⁹⁸⁶ (e i `contenuti`⁹⁸⁷) l'ombra (se a destra) va nel margine della pagina, ma il bordo e il riempimento extra vengono mantenuti all'interno dell'area del testo. Stessa cosa per le ammonizioni.

⁹⁸² <https://ctan.org/pkg/pict2e>

⁹⁸³ <https://ctan.org/pkg/ellipse>

⁹⁸⁴ <https://ctan.org/pkg/pict2e>

⁹⁸⁵ <https://ctan.org/pkg/pict2e>

⁹⁸⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

⁹⁸⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

- Le direttive `contents`⁹⁸⁸ e `topic`⁹⁸⁹ sono governate dalle stesse opzioni con il prefisso `div.` `topic`: il markup Sphinx LaTeX utilizza per entrambe le direttive lo stesso ambiente `sphinxShadowBox` che attualmente non ha ramificazioni (branch) aggiuntive, contrariamente all'ambiente `sphinxadmonition` che si ramifica secondo il nome della direttiva di ammonizione, ad es. su `sphinxnote` o su `sphinxwarning` ecc...

7.4 Macro e ambienti LaTeX

Il file del "pacchetto LaTeX" `sphinx.sty` carica vari componenti che forniscono macro di supporto (ovvero comandi) e ambienti, che vengono utilizzati nel markup prodotto sull'output del builder `latex`, prima della conversione in pdf tramite la toolchain LaTeX. Anche i file della "classe LaTeX" `sphinxhowto.cls` e `sphinxmanual.cls` definiscono o personalizzano alcuni ambienti. Tutti questi file si trovano nel repertorio (repertory) della build `latex`.

Alcuni di questi forniscono funzionalità non disponibili nei pacchetti LaTeX preesistenti e aggirano le limitazioni di LaTeX con elenchi, celle di tabelle, rendering letterale, note a piè di pagina, ecc.

Altri definiscono semplicemente macro con nomi pubblici per semplificare la sovrascrittura dei valori di default tramite contenuti aggiunti dall'utente al preambolo. Esamineremo qui la maggior parte di questi nomi pubblici, ma i valori di default devono essere esaminati nei rispettivi file di definizione.

Suggerimento: Il codice di supporto di Sphinx LaTeX è suddiviso in più file di dimensioni più piccole. Invece di aggiungere codice al preambolo tramite `latex_elements['preamble']` è anche possibile sostituire interamente uno dei file componenti del codice Sphinx LaTeX con una versione personalizzata, semplicemente includendo una copia modificata nel sorgente del progetto e aggiungendo il nome del file alla lista `latex_additional_files`. Controllare il repertorio della build LaTeX per i nomi dei file e i contenuti.

Cambiato nella versione 4.0.0: suddivisione di `sphinx.sty` in unità più piccole, per facilitare la personalizzazione di molti aspetti contemporaneamente.

Le Macro

- Comandi per lo stile del testo:

Name, maps argument #1 to:	
<code>\sphinxstrong</code>	<code>\textbf{#1}</code>
<code>\sphinxcode</code>	<code>\texttt{#1}</code>
<code>\sphinxbcode</code>	<code>\textbf{\sphinxcode{#1}}</code>

continues on next page

⁹⁸⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

⁹⁸⁹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

Tabella 1 – continua dalla pagina precedente

Name, maps argument #1 to:	
<code>\sphinxemail</code>	<code>\textsf{#1}</code>
<code>\sphinxtablecontinued</code>	<code>\textsf{#1}</code>
<code>\sphinxtitleref</code>	<code>\emph{#1}</code>
<code>\sphinxmenuselection</code>	<code>\emph{#1}</code>
<code>\sphinxguilabel</code>	<code>\emph{#1}</code>
<code>\sphinxkeyboard</code>	<code>\sphinxcode{#1}</code>
<code>\sphinxaccelerator</code>	<code>\underline{#1}</code>
<code>\sphinxcrossref</code>	<code>\emph{#1}</code>
<code>\sphinxtermref</code>	<code>\emph{#1}</code>
<code>\sphinxssamedocref</code>	<code>\emph{#1}</code>
<code>\sphinxparam</code>	<code>\emph{#1}</code>
<code>\sphinxtypeparam</code>	<code>\emph{#1}</code>
<code>\sphinxoptional</code>	[#1] con parentesi più grandi, vedere il sorgente

Added in version 1.4.5: Uso di nomi di macro con prefisso `\sphinx` per limitare le possibilità di conflitto con i pacchetti LaTeX.

Added in version 1.8: `\sphinxguilabel`

Added in version 3.0: `\sphinxkeyboard`

Added in version 6.2.0: `\sphinxparam`, `\sphinxssamedocref`

Added in version 7.1.0: `\sphinxparamcomma` che per default è una virgola seguita da uno spazio e `\sphinxparamcommaoneperline` che viene utilizzato per firme con un parametro per riga (vedere *maximum_signature_line_length*). Il default è `\texttt{,}` per rendere questi separatori di fine riga più distintivi.

Le firme (signature) delle funzioni Python vengono visualizzate come `name<space>(parameters)` o `name<space>[type parameters]<space>(parameters)` (vedere **PEP 695**⁹⁹⁰) dove la lunghezza di `<space>` è impostato su 0pt per default. Questo può essere modificato tramite `\setlength{\sphinxsignaturelistskip}{1ex}` per esempio.

- Altri stili di testo:

Name, maps argument #1 to:	
<code>\sphinxstyleindexentry</code>	<code>\texttt{#1}</code>
<code>\sphinxstyleindexextra</code>	<code>(\emph{#1})</code> (con uno spazio iniziale)
<code>\sphinxstyleindexpageref</code>	<code>, \pageref{#1}</code>
<code>\sphinxstyleindexpagemain</code>	<code>\textbf{#1}</code>
<code>\sphinxstyleindexlettergroup</code>	<code>{\Large\sffamily#1}\nopagebreak\vspace{1m}</code>
<code>\sphinxstyleindexlettergroupDefault</code>	controllare il sorgente, troppo lungo per qui
<code>\sphinxstyletopic title</code>	<code>\textbf{#1}\par\medskip</code>
<code>\sphinxstylesidebartitle</code>	<code>\textbf{#1}\par\medskip</code>

continues on next page

⁹⁹⁰ <https://peps.python.org/pep-0695/>

Tabella 2 – continua dalla pagina precedente

Name, maps argument #1 to:	
<code>\sphinxstyleothertitle</code>	<code>\textbf{#1}</code>
<code>\sphinxstyleidebarsubtitle</code>	<code>~\\textbf{#1} \smallskip</code>
<code>\sphinxstyletheadfamily</code>	<code>\sffamily</code> (<i>questo non ha argomenti</i>)
<code>\sphinxstyleemphasis</code>	<code>\emph{#1}</code>
<code>\sphinxstyleliteralempphasis</code>	<code>\emph{\sphinxcode{#1}}</code>
<code>\sphinxstylestrong</code>	<code>\textbf{#1}</code>
<code>\sphinxstyleliteralstrong</code>	<code>\sphinxcode{#1}</code>
<code>\sphinxstyleabbreviation</code>	<code>\textsc{#1}</code>
<code>\sphinxstyleliteralintitle</code>	<code>\sphinxcode{#1}</code>
<code>\sphinxstylecodecontinued</code>	<code>{\footnotesize(#1)}</code>
<code>\sphinxstylecodecontinues</code>	<code>{\footnotesize(#1)}</code>
<code>\sphinxstylenotetitle</code>	<code>\sphinxstrong{#1}<space></code>
<code>\sphinxstylehinttitle</code>	<i>idem</i>
<code>\sphinxstyleimportanttitle</code>	<i>idem</i>
<code>\sphinxstyletiptitle</code>	<i>idem</i>
<code>\sphinxstylewarningtitle</code>	<i>idem</i>
<code>\sphinxstylecautiontitle</code>	<i>idem</i>
<code>\sphinxstyleattentiontitle</code>	<i>idem</i>
<code>\sphinxstyledangerertitle</code>	<i>idem</i>
<code>\sphinxstyleerrortitle</code>	<i>idem</i>
<code>\sphinxstyleseealsotitle</code>	<code>\sphinxstrong{#1}\par\nopagebreak</code>

Added in version 1.5: Queste macro erano precedentemente 'scolpite' come `\texttt`, `\emph`, ecc., non personalizzabili.

Added in version 1.6: `\sphinxstyletheadfamily` che per default è `\sffamily` e consente più paragrafi nelle celle di intestazione delle tabelle.

Added in version 1.6.3: `\sphinxstylecodecontinued` e `\sphinxstylecodecontinues`.

Added in version 1.8: `\sphinxstyleindexlettergroup`, `\sphinxstyleindexlettergroupDefault`.

Added in version 6.2.0: `\sphinxstylenotetitle` et al. Il #1 è il nome localizzato della direttiva, con i due punti finali. Avvolgerlo come `\sphinxremovefinalcolon{#1}` se questi due punti finali devono essere rimossi. Esempi:

```
\renewcommand\sphinxstylewarningtitle[1]{%
  \underline{\textbf{\sphinxremovefinalcolon{#1}}}\par
}
\renewcommand\sphinxstylenotetitle[1]{%
  \textit{\textbf{\sphinxremovefinalcolon{#1}}}\par\nobreak
  % LaTeX syntax is complex and we would be better off using \
  \hrule.
  {\parskip0pt\noindent}%
```

(continues on next page)

(continua dalla pagina precedente)

```

\raisebox{1ex}%
  {\makebox[\linewidth]{\textcolor{sphinxnoteBorderColor}{\
→dotfill}}}}
% It is complex to obtain nice vertical spacing for both a
→paragraph
% or a list following up; this set-up is better for a paragraph
→next.
\par\vskip-\parskip
}

```

- `\sphinxtableofcontents`: Un wrapper (definito diversamente in `sphinxhowto.cls` e in `sphinxmanual.cls`) della `\tableofcontents` standard. La macro `\sphinxtableofcontentshook` viene eseguita durante la sua espansione subito prima della stessa `\tableofcontents`.

Cambiato nella versione 1.5: Precedentemente, il significato di `\tableofcontents` è stato modificato da Sphinx.

Cambiato nella versione 2.0: Le ridefinizioni hardcoded di `\l@section` e di `\l@subsection` precedentemente eseguite durante il caricamento della docclass 'manual' vengono ora eseguite successivamente tramite `\sphinxtableofcontentshook`. Questa macro viene eseguita anche dalla docclass 'howto', per default è vuota.

Suggerimento: Se si aggiunge al preambolo il caricamento del pacchetto `tocloft`, aggiungere anche al preambolo `\renewcommand\sphinxtableofcontentshook{}` altrimenti verranno ripristinati `\l@section` e `\l@subsection` annullando la personalizzazione di `tocloft`.

- `\sphinxmaketitle`: Utilizzato come impostazione di default della chiave di 'maketitle' *latex_elements*. Definita nei file di classe `sphinxmanual.cls` e `sphinxhowto.cls`.

Cambiato nella versione 1.8.3: Precedentemente, il `\maketitle` della classe di documenti LaTeX è stato modificato da Sphinx.

- `\sphinxbackoftitlepage`: Per la docclass 'manual', e se è definita, viene eseguita alla fine di `\sphinxmaketitle`, prima del `\clearpage` finale. Utilizzare la chiave 'maketitle' o la 'preamble' di *latex_elements* per aggiungere una definizione personalizzata di `\sphinxbackoftitlepage`.

Added in version 1.8.3.

- `\sphinxcite`: Un wrapper dello standard `\cite` per i riferimenti alle citazioni.

Il comando `\sphinxbox`

Added in version 6.2.0.

Il comando `\sphinxbox[key=value,...]{inline text}` è utilizzabile per "in scatolare" (box) elementi di testo in linea con tutta la personalizzazione descritta in *Chiavi aggiuntive 'sphinxsetup' simili a CSS*. È un comando LaTeX con un argomento opzionale, che è un elenco separato da virgole di coppie chiave=valore, come per *La configurazione di sphinxsetup*. Ecco l'elenco completo delle chiavi. Non usano alcun prefisso.

- `border-width`,
- `border-top-width`, `border-right-width`, `border-bottom-width`,
`border-left-width`,
- `padding`,
- `padding-top`, `padding-right`, `padding-bottom`, `padding-left`,
- `border-radius`,
- `border-top-left-radius`, `border-top-right-radius`,
`border-bottom-right-radius`, `border-bottom-left-radius`,
- `box-shadow`,
- `border-TeXcolor`, `background-TeXcolor`, `box-shadow-TeXcolor`, `TeXcolor`,
- `TeXextras`,
- e `addstrut` che è una chiave booleana, cioè da usare come `addstrut=true`, o `addstrut` da solo dove `=true` è omissso o `addstrut=false`.

Quest'ultima chiave è specifica di `\sphinxbox` e significa aggiungere un `\strut` in modo che altezze e profondità siano equalizzate tra varie istanze sulla stessa riga con contenuti variabili. Il default è `addstrut=false`.

Importante: Forse il default diventerà `addstrut=true` alla versione 7.0.0 a seconda del feedback ricevuto fino ad allora.

La combinazione `addstrut`, `padding-bottom=0pt`, `padding-top=1pt` è spesso soddisfacente.

Fare riferimento a *Chiavi aggiuntive 'sphinxsetup' simili a CSS* per importanti informazioni sulla sintassi relativa alle altre chiavi. La configurazione di default non utilizza ombre, una larghezza del bordo di `\fboxrule`, un "padding" di `\fboxsep`, angoli circolari con raggio `\fboxsep` e colori del background e dei bordi come per il rendering di default dei code-block.

Quando un utilizzo di `\sphinxbox` è nidificato all'interno di un altro, ignorerà le opzioni di quello esterno: prima reimposta tutte le opzioni al loro stato di default com'erano prima dell'applicazione delle opzioni del box esterno, poi applica quelle specifiche proprie.

È possibile modificare queste impostazioni di default tramite il comando `\sphinxboxsetup{key=value,...}`. L'effetto è cumulativo, se si usa questo comando

più volte. Qui le opzioni sono un argomento obbligatorio, quindi sono racchiuse tra parentesi graffe e non quadre.

Ecco qualche esempio di utilizzo:

```
latex_elements = {
    'preamble': r'''
% modify globally the defaults
\sphinxboxsetup{border-width=2pt,%
                border-radius=4pt,%
                background-TeXcolor=yellow!20}
% configure some styling element with some extra specific options:
\protected\def\sphinxkeyboard#1{\sphinxbox[border-TeXcolor=green]{\
→sphinxcode{#1}}}
'''
}
```

Viene fornita un'utilità `\newsphinxbox` per creare una nuova macro di boxing, ad esempio `\foo` che agirà esattamente come `\sphinxbox` ma con una determinata configurazione aggiuntiva:

```
% the specific options to \foo are within brackets
\newsphinxbox[border-radius=0pt, box-shadow=2pt 2pt]{\foo}
% then use this \foo, possibly with some extra options still:
\protected\def\sphinxguilabel#1{\foo{#1}}
\protected\def\sphinxmenuselection#1{\foo[box-shadow-TeXcolor=gray]{#1}
→}
```

I box realizzati con `\foo` obbediscono come quelli che usano direttamente `\sphinxbox` alla configurazione corrente impostata possibilmente a metà strada nel documento tramite `\sphinxboxsetup` (da un mark-up LaTeX `raw`⁹⁹¹), l'unica differenza è che hanno un set aggiuntivo iniziale di default extra.

Negli esempi precedenti, probabilmente si può usare la sintassi `\renewcommand` se la si preferisce a `\protected\def` (con `[1]` al posto di `#1` in quel caso).

Ambienti

- Una `figure`⁹⁹² può avere una legenda opzionale con elementi del corpo arbitrari: vengono realizzati in un ambiente `sphinxlegend`. La definizione di default è `\small`, e termina con `\par`.

Added in version 1.5.6: Precedentemente, `\small` era scolpito nel writer LaTeX e mancava la desinenza `\par`.

⁹⁹¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#raw>

⁹⁹² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#figure>

- Ambienti associati alle ammonizioni:

- sphinxnote,
- sphinxhint,
- sphinximportant,
- sphinxtip,
- sphinxwarning,
- sphinxcaution,
- sphinxattention,
- sphinxdanger,
- sphinxerror.

Possono essere `\renewenvironment` (*rinnovati*) individualmente e devono quindi essere definiti con un argomento (è l'intestazione dell'avviso, ad esempio **Warning**: per la direttiva `warning`⁹⁹³, se l'inglese è la lingua del documento). Le loro definizioni di default utilizzano gli ambienti `sphinxheavybox` (per gli ultimi 5) o `sphinxlightbox`, configurati per utilizzare i parametri (colori, spessore del bordo) specifici per ciascun tipo, che possono essere impostati tramite la stringa '`sphinxsetup`'.

Cambiato nella versione 1.5: Utilizzo di nomi di ambienti pubblici, personalizzazione separata dei parametri, come `noteBorderColor`, `noteborder`, `warningBgColor`, `warningBorderColor`, `warningborder`, ...

- Environment per la direttiva `seealso`: `sphinxseealso`. Richiede un argomento che sarà la stringa localizzata `See also` (*Vedi anche*) seguita da due punti.

Added in version 6.1.0.

Cambiato nella versione 6.2.0: I due punti facevano parte del markup anziché essere inseriti dall'ambiente per coerenza con il modo in cui vengono generalmente gestite le ammonizioni.

- La direttiva `contents`⁹⁹⁴ (con l'opzione `:local:`) e la direttiva `topic`⁹⁹⁵ sono implementate dall'ambiente `sphinxShadowBox`.

Added in version 1.4.2: Il codice precedente è stato rifattorizzato in un environment (*ambiente*) che consente interruzioni di pagina.

Cambiato nella versione 1.5: Opzioni `shadowsep`, `shadowsize`, `shadowrule`.

- I blocchi letterali (tramite `::` o `code-block`), sono implementati utilizzando l'ambiente `sphinxVerbatim` che è un wrapper dell'ambiente `Verbatim` dal pacchetto `fancyvrb.sty`. Aggiunge la gestione della didascalia superiore e l'andata a capo delle righe lunghe, e una cornice che consente interruzioni di pagina. All'interno delle tabelle l'ambiente utilizzato è `sphinxVerbatimintable` (non disegna una cornice, ma consente una didascalia).

⁹⁹³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#warning>

⁹⁹⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

⁹⁹⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

Cambiato nella versione 1.5: `Verbatim` mantiene esattamente lo stesso significato di `fancyvrb.sty` (anche sotto il nome `OriginalVerbatim`); all'interno delle tabelle viene utilizzato `sphinxVerbatimintable`.

Added in version 1.5: Opzioni `verbatimwithframe`, `verbatimwraplines`, `verbatimsep`, `verbatimborder`.

Added in version 1.6.6: Supporto per l'opzione `:emphasize-lines:`

Added in version 1.6.6: Personalizzazione più semplice della formattazione tramite macro LaTeX esposte all'utente come `\sphinxVerbatimHighlightLine`.

- La bibliografia utilizza `sphinxthebibliography` e l'indice del modulo Python, così come l'indice generale, utilizzano entrambi `sphinxtheindex`; questi ambienti sono wrapper degli ambienti `thebibliography` e rispettivamente `theindex` come forniti dalla classe documento (o dai pacchetti).

Cambiato nella versione 1.5: Precedentemente gli ambienti originali venivano modificati da Sphinx.

Miscellanea

- Ogni paragrafo di testo nel corpo del documento inizia con `\sphinxAtStartPar`. Attualmente, questo viene utilizzato per inserire un salto orizzontale di larghezza zero che è un trucco per consentire la sillabazione TeX della prima parola di un paragrafo in un contesto ristretto (come una cella di tabella). Per 'lualatex' che non necessita del trucco, `\sphinxAtStartPar` non fa nulla.

Added in version 3.5.0.

- Le intestazioni di sezione, sottosezione, ... vengono impostate utilizzando il comando `\titleformat` di *titlesec*.
- Per la docclass 'manual', i titoli dei capitoli possono essere personalizzati utilizzando i comandi di *fncychap*, `\ChNameVar`, `\ChNumVar`, `\ChTitleVar`. Il file `sphinx.sty` ha ridefinizioni personalizzate nel caso dell'opzione *fncychap* Bjarne.

Cambiato nella versione 1.5: Precedentemente, l'uso di *fncychap* con stili diversi da Bjarne era disfunzionale.

- Le direttive Docutils `container`⁹⁹⁶ sono supportate nell'output LaTeX: per consentire a una classe contenitore con nome `foo` di influenzare il PDF finale tramite LaTeX, è necessario solo definire nel preambolo un ambiente `sphinxclassfoo`. Un semplice esempio potrebbe essere:

```
\newenvironment{sphinxclassred}{\color{red}}{}
```

Attualmente i nomi delle classi devono contenere solo caratteri ASCII ed evitare caratteri speciali per LaTeX come `\`.

Added in version 4.1.0.

⁹⁹⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#container>

Suggerimento: Come funzionalità sperimentale, Sphinx può utilizzare file template definiti dall'utente per il sorgente LaTeX se nel progetto è presente un file denominato `_templates/latex.tex.jinja` nel progetto.

I file aggiuntivi `longtable.tex.jinja`, `tabulary.tex.jinja` e `tabular.tex.jinja` possono essere aggiunti a `_templates/` per configurare alcuni aspetti del rendering delle tabelle (come la posizione della didascalia).

Added in version 1.6: attualmente tutte le variabili del template sono instabili e non documentate.

Cambiato nella versione 7.4: Aggiunto il supporto per l'estensione `.jinja` del file, che è preferibile. I vecchi nomi di file rimangono supportati. (`latex.tex_t`, `longtable.tex_t`, `tabulary.tex_t` e `tabular.tex_t`)

Ottenere un supporto

Per domande o segnalazioni di problemi con Sphinx:

- Verificare che la domanda non esista già su [StackOverflow](https://stackoverflow.com/questions/tagged/python-sphinx)⁹⁹⁷ (con il tag `python-sphinx`)
- aprire un "topic" nella pagina [Github discussions](https://github.com/sphinx-doc/sphinx/discussions)⁹⁹⁸,
- aprire una "issue" nella pagina [Github issues](https://github.com/sphinx-doc/sphinx/issues)⁹⁹⁹,
- oppure iscriversi alla mailing list [sphinx-users](https://groups.google.com/group/sphinx-users)¹⁰⁰⁰ su Google Groups,

Esempi di altri progetti che utilizzano Sphinx si trovano nella *examples page*.

Consultare anche la guida di [ReadTheDocs](https://docs.readthedocs.io/en/stable/intro/getting-started-with-sphinx.html)¹⁰⁰¹ su come iniziare con Sphinx.

Esiste un team di traduzione in [Transifex](https://www.transifex.com/sphinx-doc/sphinx-doc/dashboard/)¹⁰⁰² di questa documentazione, grazie ai traduttori di documenti Sphinx.

⁹⁹⁷ <https://stackoverflow.com/questions/tagged/python-sphinx>

⁹⁹⁸ <https://github.com/sphinx-doc/sphinx/discussions>

⁹⁹⁹ <https://github.com/sphinx-doc/sphinx/issues>

¹⁰⁰⁰ <https://groups.google.com/group/sphinx-users>

¹⁰⁰¹ <https://docs.readthedocs.io/en/stable/intro/getting-started-with-sphinx.html>

¹⁰⁰² <https://www.transifex.com/sphinx-doc/sphinx-doc/dashboard/>

Contribuire a Sphinx

Questa guida contiene informazioni sul progetto open source Sphinx. Qui si possono trovare informazioni su come viene gestito Sphinx e scoprire come contribuire al progetto.

9.1 Contribuire a Sphinx

Ci sono molti modi in cui si può contribuire a Sphinx, sia inviando segnalazioni di bug o richieste di funzionalità, scrivendo nuova documentazione o inviando patch per comportamenti nuovi o corretti. Questa guida serve a illustrare come iniziare.

Richieste d'aiuto

La comunità Sphinx mantiene una serie di mailing list e canali IRC.

Stack Overflow col tag `python-sphinx`¹⁰⁰³

Domande e risposte sull'uso e sullo sviluppo.

sphinx-users <sphinx-users@googlegroups.com>

Mailing list per il supporto degli utenti.

sphinx-dev <sphinx-dev@googlegroups.com>

Mailing list per discussioni relative allo sviluppo.

#sphinx-doc su [irc.libera.chat](https://libera.chat)

Canale IRC per domande sullo sviluppo e supporto agli utenti.

¹⁰⁰³ <https://stackoverflow.com/questions/tagged/python-sphinx>

Bug Report e Richieste di Funzionalità

Se si riscontra un problema con Sphinx o si ha un'idea per una nuova funzionalità, la si invia al [issue tracker](#)¹⁰⁰⁴ su GitHub la si discute sulla mailing list [sphinx-dev](#)¹⁰⁰⁵.

Per le segnalazioni di bug, includere l'output prodotto durante il processo di build e anche il file di log che Sphinx crea dopo aver riscontrato un'eccezione non gestita. La posizione di questo file dovrebbe apparire verso la fine del messaggio di errore.

Includere o fornire un link ai file sorgenti coinvolti può aiutarci a risolvere il problema. Se possibile, provare a creare un progetto minimo che produca l'errore e pubblicarlo.

Contributi al codice

Il codice sorgente di Sphinx è gestito tramite Git ed è ospitato su [GitHub](#)¹⁰⁰⁶. Il modo consigliato per i nuovi contributori di inviare codice a Sphinx è quello di eseguire il fork di questo repository e inviare una pull request dopo aver confermato le modifiche al loro fork. La pull request dovrà poi essere approvata da uno degli sviluppatori principali (core) prima di essere "mergiata" nel repository principale.

Per iniziare

Prima di iniziare con una patch, consigliamo di verificare la presenza di problemi aperti o di aprire un nuovo problema per avviare una discussione su un'idea di funzionalità o su un bug. Se ci si sente a disagio o si è incerti riguardo a un problema o alle modifiche, si può inviare una e-mail alla mailing list [sphinx-dev](#).

Questi sono i passaggi fondamentali necessari per iniziare a sviluppare su Sphinx.

1. Creare un account su GitHub.
2. Effettuare il fork del repository Sphinx principale ([sphinx-doc/sphinx](#)¹⁰⁰⁷) utilizzando l'interfaccia di GitHub.
3. Clonare il repository sul proprio computer.

```
git clone https://github.com/USERNAME/sphinx
cd sphinx
```

4. Eseguire il checkout del branch appropriato.

Sphinx adotta il Semantic Versioning 2.0.0 (rif: <https://semver.org/>).

Per le modifiche che preservano la compatibilità con le versioni precedenti dell'API e delle funzionalità, dovrebbero essere incluse nella prossima versione MINOR, utilizzare il branch `A.x`.

¹⁰⁰⁴ <https://github.com/sphinx-doc/sphinx/issues>

¹⁰⁰⁵ sphinx-dev@googlegroups.com

¹⁰⁰⁶ <https://github.com/sphinx-doc/sphinx>

¹⁰⁰⁷ <https://github.com/sphinx-doc/sphinx>


```
git checkout A.x
```

Per modifiche incompatibili o sostanziali che dovrebbero attendere fino al prossimo rilascio MAJOR, utilizzare il branch `master`.

Per un rilascio urgente, un nuovo branch PATCH deve essere creato dal tag di rilascio più recente (vedere *Il processo di rilascio di Sphinx* per i dettagli).

5. Setup di un ambiente virtuale

Questo non è necessario per le unit test, grazie a `tox`, ma è necessario se per eseguire `sphinx-build` localmente o eseguire unit test senza l'aiuto di `tox`:

```
virtualenv ~/.venv  
. ~/.venv/bin/activate  
pip install -e .
```

6. Creare un nuovo branch funzionante. Scegliere un nome.

```
git checkout -b feature-xyz
```

7. Hack, hack, hack.

Scrivere il codice insieme ai test che dimostrano che il bug è stato corretto o che il tutto funziona come previsto.

8. Aggiungi un elemento all'elenco `CHANGES.rst` se la correzione o la funzionalità non è banale (piccoli aggiornamenti di documenti, correzioni di errori di battitura), poi si esegue il commit:

```
git commit -m '#42: Add useful new feature that does this.'
```

GitHub riconosce alcune frasi che possono essere utilizzate per aggiornare automaticamente il tracker dei problemi. Per esempio:

```
git commit -m 'Closes #42: Fix invalid markup in docstring of Foo.  
→ bar.'
```

chiuderebbe il problema #42.

9. "Push-are" le modifiche nel branch del repository [forked] su GitHub:

```
git push origin feature-xyz
```

10. Inviare una pull request dal proprio branch al branch corrispondente (`master` o `A.x`).

11. Attendere che uno sviluppatore "core" esamini le modifiche.

Stile della codifica

Seguire queste linee guida quando si scrive il codice per Sphinx:

- Provare a utilizzare lo stesso stile di codice utilizzato nel resto del progetto.
- Per modifiche non banali, aggiornare il file `CHANGES.rst`. Se le modifiche apportate alterano il comportamento esistente, documentarlo.
- Le nuove funzionalità dovrebbero essere documentate. Includere esempi e casi d'uso ove appropriato. Se possibile, includere un esempio di output generato.
- Quando si aggiunge una nuova variabile di configurazione, assicurarsi di documentarla e di aggiornare `sphinx/cmd/quickstart.py` se è abbastanza importante.
- Aggiungere unit test appropriati.

I controlli di stile e tipo possono essere eseguiti come segue:

```
ruff check .
mypy sphinx/
```

Le unit test

Sphinx viene testato utilizzando [pytest](https://docs.pytest.org/en/latest/)¹⁰⁰⁸ per il codice Python e [Karma](https://karma-runner.github.io)¹⁰⁰⁹ per JavaScript.

Per eseguire unit test Python, consigliamo di utilizzare `tox`, che fornisce una serie di target e consente di eseguire test su più ambienti Python diversi:

- Per elencare tutti i possibili target:

```
tox -av
```

- Per eseguire unit test per una versione specifica di Python, come Python 3.10:

```
tox -e py310
```

- Per eseguire unit test per una versione specifica di Python e attivare i warning di deprecazione in modo che vengano visualizzati nell'output del test:

```
PYTHONWARNINGS=error tox -e py310
```

- Gli argomenti a `pytest` possono essere passati tramite `tox`, ad esempio, per eseguire un test particolare:

```
tox -e py310 tests/test_module.py::test_new_feature
```

Si può anche testare installando le dipendenze nel proprio ambiente locale:

¹⁰⁰⁸ <https://docs.pytest.org/en/latest/>

¹⁰⁰⁹ <https://karma-runner.github.io>

```
pip install .[test]
```

Per eseguire test JavaScript, utilizzare npm:

```
npm install
npm run test
```

Suggerimento: karma richiede un binario Firefox da utilizzare come browser di prova.

Per i sistemi basati su Unix, è possibile specificare il percorso del binario di Firefox utilizzando:

```
FIREFOX_BIN="/Applications/Firefox.app/Contents/MacOS/firefox" npm test
```

Le nuove unit test dovrebbero essere incluse nella directory `tests` dove necessario:

- Per le correzioni di bug, si aggiunge prima un test che fallisce senza le modifiche e passa dopo la loro applicazione.
- I test che necessitano di un'esecuzione di `sphinx-build` dovrebbero essere integrati, se possibile, in uno dei moduli di test esistenti. I nuovi test che eseguono `@with_app` e poi `build_all` per qualche asserzione non sono buoni poiché *la suite di test non dovrebbe impiegare più di un minuto per essere eseguita*.

Added in version 1.8: Sphinx esegue anche test JavaScript.

Added in version 1.6: `sphinx.testing` viene aggiunto come sperimentale.

Cambiato nella versione 1.5.2: Sphinx è stato spostato da Nose a Pytest.

Da fare: Quanto segue appartiene alla guida per sviluppatori

Le funzioni di utilità e le fixture pytest per i test sono fornite in `sphinx.testing`. Uno sviluppatore di estensioni Sphinx, puoi scrivere unit test utilizzando pytest. A questo punto, `sphinx.testing` aiuterà l'implementazione del test.

Come utilizzare le fixture pytest forniti da `sphinx.testing`? Si può richiedere '`sphinx.testing.fixtures`' nei propri moduli di test o file `conftest.py` come questo:

```
pytest_plugins = 'sphinx.testing.fixtures'
```

Per conoscere un utilizzo più dettagliato, fare riferimento a `tests/conftest.py` e agli altri file `test_*.py` nella directory `tests`.

Contribuire alla documentazione

Contribuire alla documentazione implica la modifica dei file sorgente presenti nella cartella `doc/`. Per iniziare, si deve prima seguire *Per iniziare*, e poi eseguire i passaggi seguenti per lavorare con la documentazione.

Le sezioni seguenti descrivono come iniziare a contribuire con la documentazione, nonché gli aspetti chiave di alcuni strumenti diversi che utilizziamo.

Da fare: Aggiungere una guida più completa per i contributi alla documentazione.

Build della documentazione

Per creare la documentazione, eseguire il comando seguente:

```
sphinx-build -M html ./doc ./build/sphinx -W --keep-going
```

Questo analizzerà i file sorgenti della documentazione di Sphinx e genererà HTML da visualizzare in anteprima in `build/sphinx/html`.

Si può anche creare una **versione live della documentazione** visionabile nel browser. Rileverà le modifiche e ricaricherà la pagina ogni volta che si apportano modifiche. Per fare ciò, si esegue il seguente comando:

```
sphinx-autobuild ./doc ./build/sphinx/
```

Traduzioni

Le parti dei messaggi in Sphinx che vengono inserite nelle build vengono tradotte in diverse localizzazioni. Le traduzioni vengono conservate come file gettext `.po` tradotti dal template `master sphinx/locale/sphinx.pot`.

Sphinx utilizza [Babel](https://babel.pocoo.org/en/latest/)¹⁰¹⁰ per estrarre i messaggi e mantenere i file di catalogo. La directory `utils` contiene uno script di supporto, `babel_runner.py`.

- Utilizzare `python babel_runner.py extract` per aggiornare il template `.pot`.
- Usare `python babel_runner.py update` per aggiornare tutti i cataloghi esistenti delle lingue in `sphinx/locale/*/LC_MESSAGES` con i messaggi correnti nel file template.
- Usare `python babel_runner.py compile` per compilare i file `.po` in file binari `.mo` e file `.js`.

Quando viene inviato un file `.po` aggiornato, eseguire `python babel_runner.py compile` per "commit-tare" sia il catalogo sorgente che quello compilato.

¹⁰¹⁰ <https://babel.pocoo.org/en/latest/>

Quando viene inviata una nuova localizzazione, si aggiunge una nuova directory con l'identificatore della lingua ISO 639-1 e vi si inserisce il file `sphinx.po`. Non dimenticare di aggiornare i possibili valori per `language` in `doc/usage/configuration.rst`.

I messaggi fondamentali di Sphinx possono essere tradotti anche su [Transifex](https://www.transifex.com/sphinx-doc/sphinx-1/)¹⁰¹¹. Lo strumento client `tx`, fornito dal pacchetto Python `transifex_client`, è utilizzabile per estrarre traduzioni in formato `.po` da Transifex. Per farlo, si va in `sphinx/locale` e si esegue `tx pull -f -l LANG` dove `LANG` è un identificatore esistente di una lingua. È buona norma eseguire successivamente `python babel_runner.py update` per assicurarsi che il file `.po` abbia la formattazione canonica Babel.

Suggerimenti per il debug

- Eliminare la cache della build prima di creare documenti, se si apportano modifiche al codice, eseguendo il comando `make clean` o utilizzando l'opzione `sphinx-build -E`.
- Usare l'opzione `sphinx-build -P` per eseguire pdb sulle eccezioni.
- Usare `node.pformat()` e `node.asdom().toxml()` per generare una rappresentazione stampabile della struttura del documento.
- Impostare la variabile di configurazione `keep_warnings` a `True` in modo che i warning vengano visualizzati nell'output generato.
- Settare la variabile di configurazione `nitpicky` a `True` in modo che Sphinx visualizzi i riferimenti senza un target noto.
- Impostare le opzioni di debug nel [File di configurazione di Docutils](https://docutils.sourceforge.io/docs/user/config.html)¹⁰¹².
- Gli algoritmi di stemming JavaScript in `sphinx/search/non-minified-js/*.js` vengono generati utilizzando `snowball`¹⁰¹³ clonando il repository, eseguendo `make dist_libstemmer_js` e poi decomprimere il tarball generato nella directory `dist`.

I file minimizzati in `sphinx/search/minified-js/*.js` vengono generati da quelli non minimizzati utilizzando `uglifyjs` (installato tramite npm), con l'opzione `-m` per abilitare la modifica.

- I file `searchindex.js` nelle directory `tests/js/fixtures/*` vengono generati utilizzando il builder HTML standard di Sphinx sui progetti di input corrispondenti trovati in `tests/js/roots/*`. Le "fixture" forniscono dati di test utilizzati dalle unit test JavaScript di Sphinx e si possono rigenerare eseguendo lo script `utils/generate_js_fixtures.py`.

¹⁰¹¹ <https://www.transifex.com/sphinx-doc/sphinx-1/>

¹⁰¹² <https://docutils.sourceforge.io/docs/user/config.html>

¹⁰¹³ <https://github.com/snowballstem/snowball>

9.2 Il processo di rilascio di Sphinx

Controllo delle versioni

Sphinx aderisce alle versioni **PEP 440**¹⁰¹⁴, con uno schema `major.minor.micro` per il *release segment* (ad esempio 1.2.3). Le parti della versione `major`, `minor` e `micro` dovrebbero essere modificate come segue:

- La parte `major` della versione dovrebbe essere incrementata per modifiche del comportamento incompatibili e aggiornamenti dell'API pubblica.
- La parte `minor` della versione dovrebbe essere incrementata per la maggior parte delle versioni di Sphinx, dove la compatibilità con le versioni precedenti dell'API e delle funzionalità è preservata.
- La parte `micro` della versione dovrebbe essere incrementata solo per rilasci urgenti contenenti solo correzioni di bug.

Quando la parte `major` della versione viene incrementata, le parti della versione `minor` e `micro` devono essere impostate a `0`. Quando la parte `minor` della versione viene incrementata, la parte della versione `micro` deve essere impostata a `0`.

Le nuove versioni `major` dovrebbero prevedere un periodo di beta testing prima del rilascio finale.

Deprecare una funzionalità

Ci sono un paio di ragioni per cui il codice in Sphinx potrebbe essere deprecato:

- Se una funzionalità è stata migliorata o modificata in modo incompatibile con le versioni precedenti, la funzionalità o il comportamento precedente verranno deprecati.
- A volte Sphinx includerà una backport di una libreria Python che non è inclusa in una versione di Python attualmente supportata da Sphinx. Quando Sphinx non avrà più bisogno di supportare la versione precedente di Python che non include la libreria, quest'ultima verrà deprecata in Sphinx.

Come descritto nella *Politica sulla deprecazione*, la prima versione di Sphinx che deprecava una funzionalità (`A.B`) dovrebbe sollevare un `RemovedInSphinxXXWarning` (dove `XX` è la versione di Sphinx in cui la funzionalità verrà rimossa) quando viene richiamata la funzionalità deprecata. Supponendo di avere una buona copertura del test, questi warning vengono convertiti in errori quando si esegue la suite di test con i warning abilitati:

```
pytest -Wall
```

Pertanto, quando si aggiunge un `RemovedInSphinxXXWarning` è necessario eliminare o silenziare qualsiasi warning generato durante l'esecuzione dei test.

¹⁰¹⁴ <https://peps.python.org/pep-0440/>

Politica sulla deprecazione

Le versioni MAJOR e MINOR potrebbero deprecare alcune funzionalità delle versioni precedenti. Se una funzionalità è deprecata in una versione A.x, continuerà a funzionare in tutte le versioni A.x.x (per tutte le versioni di x). Continuerà a funzionare in tutte le versioni B.x.x ma genererà warning sulla deprecazione. Le funzionalità deprecate verranno rimosse nella versione C.0.0. Ciò significa che la funzionalità deprecata funzionerà almeno durante 2 versioni MAJOR.

Quindi, ad esempio, se decidessimo di iniziare la deprecazione di una funzione in Sphinx 2.x:

- Sphinx 2.x conterrà una replica compatibile con le versioni precedenti della funzione che genererà un `RemovedInSphinx40Warning`. Questa è una sottoclasse di `PendingDeprecationWarning`¹⁰¹⁵, ovvero non verrà visualizzata per default.
- Sphinx 3.x conterrà ancora la replica compatibile con le versioni precedenti, ma `RemovedInSphinx40Warning` sarà quindi una sottoclasse di `DeprecationWarning`¹⁰¹⁶ e verrà visualizzata per default.
- Sphinx 4.0 rimuoverà completamente la funzionalità.

Warning di deprecazione

Sphinx abiliterà i suoi warning `RemovedInNextVersionWarning` per default, se `PYTHONWARNINGS`¹⁰¹⁷ non è settato. Quindi si possono disabilitare con:

- `PYTHONWARNINGS= make html` (Linux/Mac)
- `export PYTHONWARNINGS=` e si esegue `make html` (Linux/Mac)
- `set PYTHONWARNINGS=` e si esegue `make html` (Windows)

Ma si possono anche abilitare esplicitamente quelli in sospeso usando ad es. `PYTHONWARNINGS=default` (vedere la [documentazione Python sulla configurazione dei warning](#)¹⁰¹⁸) per ulteriori dettagli.

Politica per il supporto della versione Python

Sphinx supporta tutte le versioni minor di Python rilasciate negli ultimi 42 mesi dalla data di rilascio prevista con un minimo di 3 versioni minor di Python. Questa politica deriva da [NEP 29](#)¹⁰¹⁹, uno standard Python di dominio scientifico.

Ad esempio, una versione di Sphinx rilasciata nel maggio 2024 supporterebbe Python 3.10, 3.11 e 3.12.

Questa è una tabella riepilogativa con la politica attuale:

¹⁰¹⁵ <https://docs.python.org/3/library/exceptions.html#PendingDeprecationWarning>

¹⁰¹⁶ <https://docs.python.org/3/library/exceptions.html#DeprecationWarning>

¹⁰¹⁷ <https://docs.python.org/3/using/cmdline.html#envvar-PYTHONWARNINGS>

¹⁰¹⁸ <https://docs.python.org/3/library/warnings.html#describing-warning-filters>

¹⁰¹⁹ https://numpy.org/neps/nep-0029-deprecation_policy.html

Data	Python
26 Dic 2021	3.8+
14 Apr 2023	3.9+
05 Apr 2024	3.10+
04 Apr 2025	3.11+
24 Apr 2026	3.12+

Procedure di rilascio

Le procedure di rilascio sono elencate in `utils/release-checklist.rst`.

9.3 Organizzazione del progetto Sphinx

La guida spiega come è organizzato il progetto Sphinx.

Sviluppatori principali

Gli sviluppatori principali (core) di Sphinx hanno accesso in scrittura al repository principale. Possono committare modifiche, accettare/rifiutare pull request e gestire elementi nell'issue tracker.

Linee guida

Di seguito sono riportate alcune linee guida generali per gli sviluppatori principali:

- Modifiche discutibili o estese dovrebbero essere inviate come pull request anziché committate direttamente al repository principale. La pull request deve essere esaminata da un altro sviluppatore principale prima di essere mergiata.
- È possibile eseguire direttamente il commit di modifiche banali, ma ci si deve assicurare di mantenere il repository in buono stato di funzionamento e che tutti i test vengano superati prima di eseguire il push delle modifiche.
- Quando si committa il codice scritto da qualcun altro, si prega di attribuire all'autore originale nel messaggio di commit e qualsiasi voce `CHANGES.rst` rilevante.

Appartenenza

L'appartenenza al "core" si basa sul continuo contributo attivo al progetto. In generale, i potenziali "core" dovrebbero dimostrare:

- una buona comprensione di uno o più componenti di Sphinx
- uno storico di contributi utili e costruttivi
- la volontà di investire tempo nel miglioramento di Sphinx

Fare riferimento a [Contribuire a Sphinx](#) per ulteriori informazioni su come iniziare.

Altri collaboratori

Non è necessario essere uno sviluppatore "core" o avere accesso in scrittura per essere coinvolto nello sviluppo di Sphinx. Si possono inviare patch o creare "pull request" da repository forked e chiedere a uno sviluppatore core di aggiungere le modifiche.

Allo stesso modo, i contributi non si limitano alle patch del codice. Accogliamo con favore anche l'aiuto nella valutazione dei bug, input sulle decisioni di progettazione, revisioni delle patch esistenti e miglioramenti della documentazione. Maggiori informazioni si trovano in [Contribuire a Sphinx](#).

Un elenco delle persone che hanno contribuito a Sphinx si trova in [Autori di Sphinx](#).

9.4 Codice di Condotta di Sphinx

Come la comunità tecnica nel suo insieme, il team e la comunità Sphinx sono composti da volontari provenienti da tutto il mondo. La diversità è un punto di forza, ma può anche portare a problemi di comunicazione e infelicità. A tal fine, abbiamo alcune regole di base a cui chiediamo alle persone di attenersi.

- **Siate amichevoli e pazienti.**
- **Siate accoglienti.** Ci impegniamo a essere una comunità che accoglie e supporta persone di ogni provenienza e identità. Ciò include, ma non è limitato a, membri di qualsiasi razza, etnia, cultura, nazione, colore, stato di immigrazione, classe sociale ed economica, livello di istruzione, sesso, orientamento sessuale, identità ed espressione di genere, età, taglia, stato di famiglia, convinzione politica, religione e capacità mentale e fisica.
- **Siate premurosi.** Il proprio lavoro verrà utilizzato da altre persone e si dipenderà dal lavoro altrui. Qualsiasi decisione presa influenzerà utenti e colleghi e si dovrebbe tenere conto di tali conseguenze nel prendere decisioni. Ricordarsi che siamo una comunità mondiale, quindi si potrebbe non comunicare nella lingua principale di qualcun altro.
- **Siate rispettosi.** Non tutti saremo sempre d'accordo, ma il disaccordo non è una scusa per un comportamento inadeguato e per le cattive maniere. Potremmo tutti provare qualche frustrazione di tanto in tanto, ma non possiamo permettere che quella frustrazione si

trasformi in un attacco personale. È importante ricordare che una comunità in cui le persone si sentono a disagio o minacciate non è produttiva. I membri della comunità Sphinx dovrebbero essere rispettosi quando trattano con altri membri e con persone esterne alla comunità Sphinx.

- **Fare attenzione alle parole che si scelgono.** Siamo una comunità di professionisti e ci comportiamo in modo professionale. Siate gentili con gli altri. Non insultare o denigrare gli altri partecipanti. Le molestie e altri comportamenti esclusivi non sono accettabili. Ciò include, ma non è limitato a:
 - Minacce o linguaggio violento diretti contro un'altra persona.
 - Scherzi e linguaggio discriminatori.
 - Pubblicare materiale sessualmente esplicito o violento.
 - Pubblicare (o minacciare di pubblicare) informazioni di identificazione personale di altre persone ("doxing").
 - Insulti personali, soprattutto quelli che utilizzano termini razzisti o sessisti.
 - Attenzioni sessuali indesiderate.
 - Sostenere o incoraggiare uno qualsiasi dei comportamenti di cui sopra.
 - Molestie ripetute nei confronti degli altri. In generale, se qualcuno chiede di fermarsi, allora ci si fermi.
- **Quando non siamo d'accordo, cercare di capirne il motivo.** I disaccordi, sia sociali che tecnici, si verificano continuamente e Sphinx non fa eccezione. È importante risolvere i disaccordi e le opinioni divergenti in modo costruttivo. Ricordarsi che siamo diversi. Persone diverse hanno visioni diverse sui problemi. Non essere in grado di capire perché qualcuno ha un punto di vista non significa che abbia torto. Non dimenticare che è umano sbagliare e incolparsi a vicenda non porta da nessuna parte. Concentrarsi invece sull'aiutare a risolvere i problemi e sull'imparare dagli errori.

Questo non è un elenco esaustivo delle cose che non si possono fare. Piuttosto, lo si prenda nello spirito con cui è inteso: una guida per facilitare l'arricchimento di tutti noi e delle comunità tecniche a cui partecipiamo.. Questo codice di condotta si applica a tutti gli spazi della comunità Sphinx.

Attribuzione

Testo originale per gentile concessione di Speak Up! progetto: <http://web.archive.org/web/20141109123859/http://speakup.io/coc.html>.

CAPITOLO 10

FAQ su Sphinx

Questo è un elenco di Frequently Asked Questions [*domande frequenti*] su Sphinx. Sentitevi liberi di suggerire nuove domande!

10.1 Come si fa per...

... creare file PDF senza LaTeX?

[rinohtype](#)¹⁰²⁰ fornisce un builder PDF utilizzabile come sostituto immediato del builder LaTeX.

... ricavare i numeri delle sezioni?

Sono automatici nell'output LaTeX; per l'HTML, si fornisce un'opzione `:numbered:` alla direttiva `toctree` da cui si vuole iniziare la numerazione.

... personalizzare l'aspetto dei file HTML creati?

Usare i temi, vedere [Temi HTML](#).

... aggiungere sostituzioni o inclusioni globali?

Si aggiungono nel valore di configurazione `rst_prolog` o `rst_epilog`.

... visualizzare l'intero albero del sommario nella barra laterale?

Si usa il richiamabile `toctree` in un template di layout personalizzato, probabilmente nel blocco `sidebartoc`.

... scrivere la propria estensione?

Consultare [Tutorial](#).

... convertire dai propri documenti esistenti utilizzando il markup MoinMoin?

Il modo più semplice è convertire in xhtml, poi convertire [xhtml in reST](#)¹⁰²¹. Si dovranno

¹⁰²⁰ <https://github.com/brecht/rinohtype>

¹⁰²¹ <https://docutils.sourceforge.io/sandbox/xhtml2rest/xhtml2rest.py>

comunque contrassegnare le classi e simili, ma i titoli e gli esempi di codice vengono visualizzati in modo pulito.

Per molte altre estensioni e altri contributi, vedere il repository [sphinx-contrib](#)¹⁰²².

10.2 Usare Sphinx con...

Read the Docs

[Read the Docs](#)¹⁰²³ è un servizio di hosting di documentazione basato su Sphinx. Ospiterà la documentazione di Sphinx, oltre a supportare una serie di altre funzionalità tra cui il supporto della versione, la generazione di PDF e altro ancora. La guida [Getting Started](#)¹⁰²⁴ è un buon punto di partenza.

Epydoc

siste un'estensione di terze parti che fornisce un [api role](#)¹⁰²⁵ che fa riferimento ai documenti API di Epydoc per un determinato identificatore.

Doxygen

Michael Jones sta sviluppando un bridge [*ponte*] reST/Sphinx verso doxygen chiamato [breathe](#)¹⁰²⁶.

SCons

Glenn Hutchings ha scritto uno script di build SCons per la build della documentazione Sphinx; è ospitato qui: <https://bitbucket-archive.softwareheritage.org/projects/zo/zondo/sphinx-scons.html>

PyPI

Jannis Leidel ha scritto un [comando setuptools](#)¹⁰²⁷ che carica automaticamente la documentazione Sphinx nell'area della documentazione del pacchetto PyPI su <https://pythonhosted.org/>.

GitHub Pages

Aggiungere [sphinx.ext.githubpages](#) al progetto. Consente di pubblicare il documento in GitHub Pages. Genera automatica i file di supporto (helper) per GitHub Pages durante la build dei documenti HTML.

MediaWiki

Vedere [sphinx-wiki](#)¹⁰²⁸, un progetto di Kevin Dunn.

Google Analytics

Si può utilizzare un template custom layout.html, come questo:

¹⁰²² <https://github.com/sphinx-contrib/>

¹⁰²³ <https://readthedocs.org>

¹⁰²⁴ <https://docs.readthedocs.io/en/stable/intro/getting-started-with-sphinx.html>

¹⁰²⁵ <https://git.savannah.gnu.org/cgit/kenozoid.git/tree/doc/extapi.py>

¹⁰²⁶ <https://github.com/michaeljones/breathe/tree/master>

¹⁰²⁷ <https://pypi.org/project/Sphinx-PyPI-upload/>

¹⁰²⁸ <https://bitbucket-archive.softwareheritage.org/projects/ke/kevindunn/sphinx-wiki.html>

```

{% extends "!layout.html" %}

{%- block extrahead %}
{{ super() }}
<script>
    var _gaq = _gaq || [];
    _gaq.push(['_setAccount', 'XXX account number XXX']);
    _gaq.push(['_trackPageview']);
</script>
{% endblock %}

{% block footer %}
{{ super() }}
<div class="footer">This page uses <a href="https://analytics.
→google.com/">
Google Analytics</a> to collect statistics. You can disable it by_
→blocking
the JavaScript coming from www.google-analytics.com.
<script>
    (function() {
        var ga = document.createElement('script');
        ga.src = ('https:' == document.location.protocol ?
            'https://ssl' : 'https://www') + '.google-analytics.
→com/ga.js';
        ga.setAttribute('async', 'true');
        document.documentElement.firstChild.appendChild(ga);
    })();
</script>
</div>
{% endblock %}

```

Google Search

Per sostituire la funzione di ricerca nativa di Sphinx con Google Search, si procede come segue:

1. Si va su <https://cse.google.com/cse/all> per creare lo snippet di codice di Google Search.
2. Copiare lo snippet di codice e incollarlo in `_templates/searchbox.html` nel progetto Sphinx:

```

<div>
  <h3>{{ _('Quick search') }}</h3>
  <script>
    (function() {
        var cx = '.....';
        var gcse = document.createElement('script');
        gcse.async = true;

```

(continues on next page)

(continua dalla pagina precedente)

```
gcse.src = 'https://cse.google.com/cse.js?cx=' + cx;
var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(gcse, s);
})();
</script>
<gcse:search></gcse:search>
</div>
```

3. Aggiungere `searchbox.html` al valore di configurazione `html_sidebars`.

10.3 Sphinx e Docutils

tl;dr: *docutils* converte *reStructuredText* in più formati di output. Sphinx si basa su *docutils* per consentire la build di corpi di documentazione con riferimenti incrociati e indicizzati.

*docutils*¹⁰²⁹ è un sistema di elaborazione testi per convertire la documentazione in testo semplice in altri formati più ricchi. Come notato nella *documentazione docutils*¹⁰³⁰, *docutils* usa i *reader* per leggere un documento, i *parser* per analizzare formati di testo semplice in una rappresentazione ad albero interna composta da diversi tipi di *nodi* e *writer* [scrittori] per produrre questo albero in vari formati di documenti. *docutils* fornisce parser per un formato di testo semplice - *reStructuredText*¹⁰³¹ - oltre ad altri, sono stati implementati parser *out-of-tree* incluso il parser *Markdown parser* di Sphinx. D'altra parte, fornisce "writer" per molti formati diversi tra cui HTML, LaTeX, pagine man, Open Document Format e XML.

docutils espone tutte le sue funzionalità attraverso una varietà di *tool front-end*¹⁰³², come *rst2html*, *rst2odt* e *rst2xml*. Fondamentalmente, però, tutti questi strumenti, e lo stesso *docutils*, riguardano documenti singoli. Non supportano concetti come i riferimenti incrociati, l'indicizzazione di documenti o la costruzione di una gerarchia di documenti (che in genere si manifesta in un sommario).

Sphinx si basa su *docutils* sfruttando i lettori e i parser di *docutils* e fornendo il proprio *Builder*. Di conseguenza, Sphinx avvolge alcuni dei *writer* forniti da *docutils*. Ciò consente a Sphinx di fornire molte funzionalità che semplicemente non sarebbero possibili con i *docutils*, come quelle descritte sopra.

¹⁰²⁹ <https://docutils.sourceforge.io/>

¹⁰³⁰ <https://docutils.sourceforge.io/docs/dev/hacking.html>

¹⁰³¹ <https://docutils.sourceforge.io/rst.html>

¹⁰³² <https://docutils.sourceforge.io/docs/user/tools.html>

10.4 Info Epub

L'elenco seguente fornisce alcuni suggerimenti per la creazione di file epub:

- Dividere il testo in più file. Quanto più lunghi sono i singoli file HTML, tanto più tempo impiega il reader di ebook per realizzarli. In casi estremi, il rendering può richiedere fino a un minuto.
- Provare a ridurre al minimo il markup. Questo ripaga anche in termini di tempo di rendering.
- Per alcuni reader è possibile utilizzare caratteri nativi o esterni utilizzando la direttiva CSS `@font-face`. Questo è *estremamente* utile per i listati di codici che spesso vengono tagliati al margine destro. Il carattere di default Courier (o variante) è piuttosto ampio e si possono visualizzare solo fino a 60 caratteri su una riga. Se lo si sostituisce con un font più stretto, si possono ottenere più caratteri su una riga. Si può utilizzare anche [FontForge](https://fontforge.github.io/)¹⁰³³ e creare varianti ristrette di alcuni font gratuiti. Nel mio caso ottengo fino a 70 caratteri su una riga.

Potrebbe essere necessario sperimentare un po' finché non si ottengono risultati ragionevoli.

- Testare gli epub creati. Si possono utilizzare diverse alternative. Quelli di cui sono a conoscenza sono [Epubcheck](https://github.com/IDPF/epubcheck)¹⁰³⁴, [Calibre](https://calibre-ebook.com/)¹⁰³⁵, [FBreader](https://fbreader.org/)¹⁰³⁶ (anche se non esegue il rendering del CSS) e [Bookworm](https://www.oreilly.com/bookworm/index.html)¹⁰³⁷. Per Bookworm, si può scaricare il codice sorgente da <https://code.google.com/archive/p/threepress> ed eseguire il proprio server locale.
- I div "floating" di grandi dimensioni non vengono visualizzati correttamente. Se coprono più di una pagina, il div viene mostrato solo sulla prima pagina. In tal caso si può copiare `epub.css` dalla directory `sphinx/themes/epub/static/` alla propria directory locale `_static/` e rimuovere le impostazioni "float".
- I file inseriti al di fuori della direttiva `toctree` devono essere inclusi manualmente. Questo a volte si applica alle appendici, ad es. il glossario e gli indici. Si possono aggiungere con l'opzione `epub_post_files`.
- La gestione della copertina dell'epub differisce dalla procedura `reStructuredText` che risolve automaticamente i path delle immagini e inserisce le immagini nella directory `_images`. Per la copertina dell'epub, si inserisce l'immagine nella directory `html_static_path` e vi si fa riferimento con il suo percorso completo nell'opzione di configurazione `epub_cover`.
- Il comando `kindlegen`¹⁰³⁸ può convertire dal file risultante `epub3` al file `.mobi` per Kindle. Si può ottenere `yourdoc.mobi` sotto `_build/epub` dopo il seguente comando:

¹⁰³³ <https://fontforge.github.io/>

¹⁰³⁴ <https://github.com/IDPF/epubcheck>

¹⁰³⁵ <https://calibre-ebook.com/>

¹⁰³⁶ <https://fbreader.org/>

¹⁰³⁷ <https://www.oreilly.com/bookworm/index.html>

¹⁰³⁸ <https://www.amazon.com/gp/feature.html?docId=1000765211>

```
$ make epub
$ kindlegen _build/epub/yourdoc.epub
```

Il comando `kindlegen` non accetta documenti che hanno titoli di sezione che circondano la direttiva `toctree`:

```
Section Title
=====

.. toctree::

    subdocument

Section After Toc Tree
=====
```

`kindlegen` presuppone che tutti i documenti siano ordinati in linea, ma il documento risultante ha un ordine complicato per `kindlegen`:

```
`parent.xhtml` -> `child.xhtml` -> `parent.xhtml`
```

Se si riceve il seguente errore, correggere la struttura del documento:

```
Error(prcgen):E24011: TOC section scope is not included in the
↳parent chapter:(title)
Error(prcgen):E24001: The table of content could not be built.
```

10.5 Info su Texinfo

Esistono due programmi principali per leggere i file Info, `info` e Emacs GNU. Il programma `info` ha meno funzionalità ma è disponibile nella maggior parte degli ambienti Unix ed è rapidamente accessibile dal terminale. Emacs fornisce una migliore visualizzazione dei caratteri e dei colori e supporta un'ampia personalizzazione (ovviamente).

Visualizzazione dei Link

Un problema evidente che si potrebbe riscontrare con i file Info generati è il modo in cui vengono visualizzati i riferimenti. Se si legge il sorgente di un file Info, un riferimento a questa sezione sarebbe simile a:

```
* note Displaying Links: target-id
```

Nel lettore stand-alone, `info`, i riferimenti vengono visualizzati così come appaiono nel sorgente. Emacs, d'altro canto, sostituirà per default `*note:` con `see` e nasconderà il `target-id`. Per esempio:

Visualizzazione dei Link

È possibile disabilitare la generazione dei riferimenti in linea in un documento con `texinfo_cross_references`. Ciò rende un file info più leggibile con lo stand-alone (info).

Il comportamento esatto di come Emacs visualizza i riferimenti dipende dalla variabile `Info-hide-note-references`. Se impostato sul valore `hide`, Emacs nasconderà sia la parte `*note:` che il `target-id`. Questo è generalmente il modo migliore per visualizzare i documenti basati su Sphinx poiché spesso fanno un uso frequente di link e non tengono conto di questa limitazione. Tuttavia, la modifica di questa variabile influisce sul modo in cui vengono visualizzati tutti i documenti Info e la maggior parte tiene conto di questo comportamento.

Se si vuole che Emacs visualizzi i file Info prodotti da Sphinx usando il valore `hide` per `Info-hide-note-references` e il valore di default per tutti gli altri file Info, provare ad aggiungere il seguente codice Emacs Lisp al file di start-up, `~/.emacs.d/init.el`.

```
(defadvice info-insert-file-contents (after
                                       sphinx-info-insert-file-contents
                                       activate)
  "Hack to make `Info-hide-note-references' buffer-local and
  automatically set to `hide' iff it can be determined that this file
  was created from a Texinfo file generated by Docutils or Sphinx."
  (set (make-local-variable 'Info-hide-note-references)
       (default-value 'Info-hide-note-references))
  (save-excursion
    (save-restriction
      (widen) (goto-char (point-min))
      (when (re-search-forward
              "^Generated by \\(Sphinx\\|Docutils\\)"
              (save-excursion (search-forward "\x1f" nil t)) t)
        (set (make-local-variable 'Info-hide-note-references)
              'hide))))))
```

Note

Le seguenti note possono essere utili per creare file Texinfo:

- Ogni sezione corrisponde a un diverso nodo nel file Info.
- Non è possibile eseguire correttamente l'escape dei due punti (:) nelle voci di menu e negli xref. Verranno sostituiti con il punto e virgola (;).
- I link a file Info esterni possono essere creati utilizzando lo schema URI alquanto ufficiale `info:Texinfo#makeinfo_options`. Per esempio:

```
info:Texinfo#makeinfo_options
```


11.1 Manutentori

Elencati in ordine alfabetico per nome, cognome

- Adam Turner <@AA-Turner>
- B         Tran <@picnixz>
- Chris Sewell <@chrisjsewell>
- Fran       Freitag <@francoisfreitag>
- Jakob Lykke Andersen <@jakobandersen>
- Jean-Fran       Burnol <@jfbu>
- Stephen Finucane <@stephenfin>
- Takayuki Shimizukawa <@shimizukawa>
- Takeshi Komiya <@tk0miya>

11.2 Contributori

Elencati in ordine alfabetico per nome, cognome

- Adri       Chaves (Gallaecio) -- miglioramenti del coverage builder
- Alastair Houghton -- Builder Apple Help
- Alexander Todorov -- test e miglioramenti di inheritance_diagram
- Andi Albrecht -- tema agogo

- Antonio Valentino -- builder qthelp, docstring inheritance
- Antti Kaihola -- estensione di doctest (opzione skipif)
- Barry Warsaw -- miglioramenti al comando di setup
- Ben Egan -- miglioramenti di Napoleon
- Benjamin Peterson -- unittest
- Blaise Laflamme -- thema pyramid
- Bruce Mitchener -- miglione secondarie a epub
- Buck Evan -- builder dummy
- Charles Duffy -- estensione graphviz originale
- Chris Lamb -- correzioni alla riproducibilità
- Christopher Perkins -- integrazione dell'autosummary
- Dan MacKinlay -- correzioni ai metadati
- Daniel Bültmann -- l'estensione todo
- Daniel Neuhäuser -- dominio JavaScript, supporto a Python 3 (GSOC)
- Daniel Pizetta -- miglioramenti al diagramma dell'ereditarietà
- Dave Kuhlman -- il writer originale di LaTeX
- Doug Hellmann -- miglioramenti a graphviz
- Eric N. Vander Weele -- miglioramenti ad autodoc
- Etienne Desautels -- modulo apidoc
- Ezio Melotti -- barra laterale collassabile
- Filip Vavera -- direttiva napoleon di todo
- Glenn Matthews -- miglioramenti alla signature del dominio python
- Gregory Szorc -- miglioramenti delle prestazioni
- Henrique Bastos -- supporto SVG per l'estensione graphviz
- Hernan Grecco -- miglioramenti della ricerca
- Hong Xu -- supporto svg nell'estensione imgmath e varie correzioni di bug
- Horst Gutmann -- supporto all'internazionalizzazione
- Hugo van Kemenade -- supporto per FORCE_COLOR e NO_COLOR
- Ian Lee -- miglioramenti a quickstart
- Jacob Mason -- libreria websupport (progetto GSOC)
- Jeppe Pihl -- miglioramenti a literalinclude
- Joel Wurtz -- supporto cellspanning in LaTeX

- John Waltman -- Builder Texinfo
- Josip Dzlonga -- builder coverage
- Julien Palard -- Colspan e rowspan nel builder text
- Kevin Dunn -- estensione MathJax
- KINEBUCHI Tomohiko -- tipizzazione di Sphinx e di docutils
- Kurt McKee -- aggiornamenti alla documentazione
- Lars Hupfeldt Nielsen - correzione della modalità md5 di OpenSSL FIPS
- Łukasz Langa -- supporto parziale per autodoc
- Marco Buttu -- estensione doctest (opzione pyversion)
- Martin Hans -- miglioramenti ad autodoc
- Martin Larralde -- ulteriori ammonizioni napoleon
- Martin Mahner -- tema nature
- Matthew Fernandez -- correzione all'estensione todo
- Matthew Woodcraft -- miglioramenti all'output text
- Michael Droettboom -- estensione inheritance_diagram
- Michael Wilson -- Supporto a Intersphinx HTTP basic auth
- Nathan Damon -- bugfix nella validazione di path statici nei builder html
- Pauli Virtanen -- miglioramenti ad autodoc, estensione autosummary
- Rob Ruana -- estensione napoleon
- Robert Lehmann -- builder gettext (progetto GSOC)
- Roland Meister -- builder di epub
- Sebastian Wiesner -- gestione dell'immagine [image], supporto di distutils
- Stefan Seefeld -- miglioramenti a toctree
- Stefan van der Walt -- estensione autosummary
- T. Powers -- miglioramenti all'output HTML
- Taku Shimizu -- builder epub3
- Thomas Lamb -- builder linkcheck
- Thomas Waldmann -- correzioni al modulo apidoc
- Tim Hoffmann -- miglioramenti al tema
- Vince Salvino -- miglioramenti a alla ricerca JavaScript
- Will Maier -- builder della directory HTML
- Zac Hatfield-Dodds -- miglioramenti a al reporting doctest, prestazioni di intersphinx

11.3 Ex manutentori

Elencati in ordine alfabetico per nome, cognome

Gli ex manutentori sono coloro che non si sono impegnati negli ultimi due anni. Coloro che sono nell'elenco seguente possono diventare nuovamente manutentori attivi in qualsiasi momento.

- Armin Ronacher <@mitsuhiko>
- Daniel Neuhäuser <@DasIch>
- Georg Brandl <georg@python.org>
- Rob Ruana <@RobRuana>
- Robert Lehmann <@lehmannro>
- Timotheus Kampik <@TimKam>
- Yoshiki Shibukawa <@shibukawa>

Molte grazie per tutti i contributi!

Strumenti da riga di comando

Queste sono le applicazioni fornite nativamente in Sphinx.

12.1 Applicazioni Principali

sphinx-quickstart

Sinossi

sphinx-quickstart

Descrizione

sphinx-quickstart è uno strumento interattivo che pone alcune domande sul progetto e poi genera una directory di documentazione completa e un Makefile di esempio da utilizzare con *sphinx-build(1)*.

Opzioni

-q, --quiet

Modalità silenziosa che ignora la procedura guidata interattiva per specificare le opzioni. Questa opzione richiede le opzioni *-p*, *-a* e *-v*.

-h, --help, --version

Visualizza il riepilogo dell'utilizzo o la versione di Sphinx.

Opzioni della Struttura

--sep

Se specificato, separa le directory sorgenti e di build.

--no-sep

Se specificato, crea la directory di build nella directory sorgente.

--dot=DOT

All'interno della directory root verranno create altre due directory; "_templates" per i template HTML personalizzati e "_static" per gli stylesheet personalizzati e altri file statici. È possibile inserire un altro prefisso (ad esempio ".") per sostituire l'underscore.

Opzioni di Base del Progetto

-p PROJECT, --project=PROJECT

Verrà impostato il nome del progetto. (vedere [project](#)).

-a AUTHOR, --author=AUTHOR

I nomi degli autori. (vedere [copyright](#)).

-v VERSION

La versione del progetto. (vedere [version](#)).

-r RELEASE, --release=RELEASE

La release del progetto. (vedere [release](#)).

-l LANGUAGE, --language=LANGUAGE

Lingua del documento. (vedere [language](#)).

--suffix=SUFFIX

Suffisso dei file sorgenti. (vedere [source_suffix](#)).

--master=MASTER

Nome del documento principale. (vedere [root_doc](#)).

Opzioni dell'Estensione

--ext-autodoc

Abilita l'estensione [sphinx.ext.autodoc](#).

--ext-doctest

Abilita l'estensione [sphinx.ext.doctest](#).

--ext-intersphinx

Abilita l'estensione [sphinx.ext.intersphinx](#).

--ext-todo

Abilita l'estensione [sphinx.ext.todo](#).

--ext-coverage

Abilita l'estensione *sphinx.ext.coverage*.

--ext-imgmath

Abilita l'estensione *sphinx.ext.imgmath*.

--ext-mathjax

Abilita l'estensione *sphinx.ext.mathjax*.

--ext-ifconfig

Abilita l'estensione *sphinx.ext.ifconfig*.

--ext-viewcode

Abilita l'estensione *sphinx.ext.viewcode*.

--ext-githubpages

Abilita l'estensione *sphinx.ext.githubpages*.

--extensions=EXTENSIONS

Abilita estensioni arbitrarie.

Opzioni per la Creazione di Makefile e Batchfile

--use-make-mode (-m), --no-use-make-mode (-M)

Makefile/make.bat usa (o non usa) *make-mode*. Il default è use, che genera un Makefile/make.bat più conciso.

Cambiato nella versione 1.5: Il default è make-mode.

Cambiato nella versione 7.3: Il supporto per disabilitare la modalità make-mode verrà rimosso in Sphinx 8.

--makefile, --no-makefile

Crea (o non crea) makefile.

--batchfile, --no-batchfile

Crea (o non crea) file batch

Template di progetto

Added in version 1.5: Opzioni per i template di progetto per sphinx-quickstart

-t, --templatedir=TEMPLATEDIR

La directory per i file template. Si possono modificare i template dei file di progetto sphinx generati da quickstart. Sono consentiti i seguenti file template Jinja2:

- root_doc.rst_t
- conf.py_t
- Makefile_t

- `Makefile.new_t`
- `make.bat_t`
- `make.bat.new_t`

Nel dettaglio, fare riferimento ai file template di sistema forniti da Sphinx. (`sphinx/templates/quickstart`)

-d NAME=VALUE

Definisce una variabile template

Vedere anche

sphinx-build(1)

sphinx-build

Sinossi

sphinx-build [*options*] < sourcedir > < outputdir > [*filenames ...*]

Descrizione

sphinx-build genera la documentazione dai file in < sourcedir > e la inserisce in < outputdir >.

sphinx-build cerca < sourcedir >/conf.py per le impostazioni di configurazione. *sphinx-quickstart(1)* è utilizzabile per generare file template, incluso conf.py.

sphinx-build può creare documentazione in diversi formati. Un formato viene selezionato specificando il nome del builder sulla riga di comando; per default è HTML. I builder possono anche eseguire altre attività relative all'elaborazione della documentazione. Per un elenco dei builder disponibili, fare riferimento a *I Builder*.

Per default, tutto ciò che è obsoleto [outdated] viene creato. L'output per i soli file selezionati può essere creato specificando i singoli nomi dei file.

Opzioni

-M buildername

Seleziona un builder, utilizzando il *make-mode*. Consultare *I Builder* per un elenco di tutti i builder nativi di Sphinx. Le estensioni possono aggiungere i propri builder.

Importante: Sphinx riconosce l'opzione **-M** solo se viene utilizzata per prima, insieme alle directory sorgenti e di output, prima che venga passata qualsiasi altra opzione. Per esempio:

```
sphinx-build -M html ./source ./build -W --keep-going
```

Il *make-mode* fornisce la stessa funzionalità di build di un *Makefile* o *Make.bat* di default e fornisce le seguenti pipeline di build aggiuntive:

latexpdf

Build dei file LaTeX e loro esecuzione con **pdflatex**, o secondo l'impostazione di *latex_engine*. Se *language* è impostato su 'ja', utilizzerà automaticamente la pipeline **platex/dvipdfmx** da latex a PDF.

info

Build dei file Texinfo e loro esecuzione con **makeinfo**.

Nota: Le posizioni di default delle directory di output quando si utilizza *make-mode* differiscono da quelle di default quando si utilizza *-b*.

- I doctree vengono salvati in <outputdir>/doctrees
 - i file di output vengono salvati in <outputdir>/<builder name>
-

Added in version 1.2.1.

-b buildername, --builder buildername

Selezione di un builder.

Consultare *I Builder* per un elenco di tutti i builder nativi di Sphinx. Le estensioni possono aggiungere i propri builder.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa *--builder*.

-a, --write-all

Se fornito, scrive sempre tutti i file di output. Il default prevede la scrittura dei file di output solo per i file sorgenti nuovi o modificati. (Questo potrebbe non applicarsi a tutti i builder).

Nota: Questa opzione non rilegge i file di sorgente. Per leggere e rielaborare tutti i file, utilizzare invece *--fresh-env*.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa *--write-all*.

-E, --fresh-env

Non utilizzare un *environment* salvato (la struttura che memorizza nella cache tutti i riferimenti incrociati), ma ricostruirlo completamente. Il default prevede la lettura e l'analisi solo dei file sorgenti nuovi o modificati dall'ultima esecuzione.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa *--fresh-env*.

-t tag, --tag tag

Definisce il tag *tag*. Questo è rilevante per le direttive *only* che includono il loro contenuto "solo" se questo tag è impostato.

Added in version 0.6.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--tag`.

-d path, --doctree-dir path

Poiché Sphinx deve leggere e analizzare tutti i file sorgenti prima di poter scrivere un file di output, quelli analizzati vengono memorizzati nella cache come "doctree pickles". Normalmente, questi file vengono inseriti in una directory chiamata `.doctrees` nella directory di build; con questa opzione si può selezionare una directory cache diversa (i doctree possono essere condivisi tra tutti i builder).

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--doctree-dir`.

-j N, --jobs N

Distribuisce la compilazione su *N* processi in parallelo, per rendere più efficace la compilazione su macchine multiprocessore. Notare che non tutte le parti e non tutti i builder di Sphinx possono essere parallelizzati. Se viene fornito l'argomento `auto`, Sphinx usa il numero di CPU come *N*. Il default è 1.

Added in version 1.2: Questa opzione dovrebbe essere considerata *sperimentale*.

Cambiato nella versione 1.7: Supporto dell'argomento `auto`.

Cambiato nella versione 6.2: Aggiunta l'opzione estesa `--jobs`.

-c path, --config-dir path

Non cerca `conf.py` nella directory sorgente, ma utilizza invece la directory di configurazione fornita. Notare che è previsto che vari altri file e path forniti dai valori di configurazione siano relativi alla directory di configurazione, quindi dovranno essere presenti anche in tale posizione.

Added in version 0.3.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--config-dir`.

-C, --isolated

Non cerca un file di configurazione; accetta le opzioni solo tramite l'opzione `--define`.

Added in version 0.5.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--isolated`.

-D setting=value, --define setting=value

Sovrascrive un valore di configurazione impostato nel file `conf.py`. Il valore deve essere un numero, una stringa, una lista o un valore di dizionario.

Per le liste, si possono separare gli elementi con una virgola come ad esempio: `-D html_theme_path=path1,path2`.

Per i valori di dizionario, fornire il nome dell'impostazione e la chiave in questo modo: `-D latex_elements.docclass=scrartcl`.

Per i valori booleani, si usa 0 o 1 come valore.

Cambiato nella versione 0.6: Il valore può ora essere un valore di dizionario.

Cambiato nella versione 1.3: Il valore ora può anche essere una lista.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--define`.

-A name=value, --html-define name=value

Assegna il *name* al *value* nei template HTML.

Added in version 0.5.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--html-define`.

-n, --nitpicky

Esegue in modalità nit-picky (pignola). Attualmente, questo genera warning per tutti i riferimenti mancanti. Vedere la configurazione di `nitpick_ignore` per avere un modo per escludere alcuni riferimenti come "known missing" (*notoriamente mancanti*).

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--nitpicky`.

-N, --no-color

Non emettere output colorato.

Cambiato nella versione 1.6: Aggiunta l'opzione estesa `--no-color`.

--color

Emettere output colorato. Rilevato automaticamente per default.

Added in version 1.6.

-v, --verbose

Aumentare la verbosità (a livello di log). Questa opzione può essere utilizzata fino a tre volte per ottenere più output nel log del debug. Implica `-T`.

Added in version 1.2.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--verbose`.

-q, --quiet

Non emette nulla sull'output standard, scrivere solo avvisi ed errori nello "standard error".

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--quiet`.

-Q, --silent

Non emette nulla sull'output standard, inoltre sopprime i warning. Solo gli errori vengono scritti nello "standard error".

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--silent`.

-w file, --warning-file file

Scriva i warning (e gli errori) nel file specificato, oltre che nell'o "standard error".

Cambiato nella versione 7.3: Le sequenze di controllo ANSI vengono rimosse durante la scrittura su *file*.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--warning-file`.

-W, --fail-on-warning

Trasforma i warning in errori. Ciò significa che la build si ferma al primo warning e `sphinx-build` esce con lo stato 1.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--fail-on-warning`.

--keep-going

Con l'opzione `-W`, continua l'elaborazione quando riceve warning sulla fine della build e `sphinx-build` esce con lo stato 1.

Added in version 1.8.

-T, --show-traceback

Visualizza il traceback completo quando si verifica un'eccezione non gestita. Altrimenti, viene visualizzato solo un riepilogo e le informazioni di traceback vengono salvate in un file per ulteriori analisi.

Added in version 1.2.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--show-traceback`.

-P, --pdb

(Utile solo per il debugging). Eseguire il debugger Python, `pdb`¹⁰³⁹, se si verifica un'eccezione non gestita durante la build.

Cambiato nella versione 7.3: Aggiunta l'opzione estesa `--pdb`.

-h, --help, --version

Visualizza il riepilogo dell'utilizzo o la versione di Sphinx.

Added in version 1.2.

Si possono anche fornire uno o più nomi di file sulla riga di comando dopo le directory sorgente e di build. Sphinx proverà poi a creare solo questi file di output (e le relative dipendenze).

Variabili d'Ambiente

sphinx-build fa riferimento alle seguenti variabili di ambiente:

MAKE

Un path per dare il comando. È consentito anche un nome di comando. **sphinx-build** lo usa per invocare il processo di sub-build in make-mode.

¹⁰³⁹ <https://docs.python.org/3/library/pdb.html#module-pdb>

Opzioni di Makefile

I file `Makefile` e `make.bat` creati da **sphinx-quickstart** di solito eseguono **sphinx-build** solo con le opzioni `-b` e `-d`. Tuttavia, supportano le seguenti variabili per personalizzarne il comportamento:

PAPER

Questo imposta la chiave 'papersize' di *latex_elements*: cioè `PAPER=a4` la imposta su 'a4paper' e `PAPER=letter` su 'letterpaper'.

Nota: L'utilizzo di questa variabile d'ambiente è stato interrotto in Sphinx 1.5 poiché `a4` e `letter` sono diventati un'opzione per il documento LaTeX al posto del necessario `a4paper`, risp. `letterpaper`. Corretto nella 1.7.7.

SPHINXBUILD

Il comando da utilizzare al posto di `sphinx-build`.

BUILDDIR

La directory di build da utilizzare al posto di quella scelta in **sphinx-quickstart**.

SPHINXOPTS

Opzioni aggiuntive per **sphinx-build**. Queste opzioni possono essere impostate anche tramite la variabile abbreviata **O** ('o' maiuscolo).

NO_COLOR

Quando impostata (indipendentemente dal valore), **sphinx-build** non utilizzerà il colore nell'output del terminale. `NO_COLOR` ha la precedenza su `FORCE_COLOR`. Vedere no-color.org¹⁰⁴⁰ per altre librerie che supportano questo standard della community.

Added in version 4.5.0.

FORCE_COLOR

Quando impostato (indipendentemente dal valore), **sphinx-build** utilizzerà il colore nell'output del terminale. `NO_COLOR` ha la precedenza su `FORCE_COLOR`.

Added in version 4.5.0.

Deprecazione dei Warning

Se vengono visualizzati avvisi di deprecazione come `RemovedInSphinxXXXWarning` durante la build del documento di un utente, alcune estensioni Sphinx utilizzano funzionalità deprecate. In tal caso, si devono segnalare all'autore dell'estensione.

Per disabilitare i warning di deprecazione, si imposta la variabile d'ambiente `PYTHONWARNINGS=` nell'environment. Per esempio:

- `PYTHONWARNINGS= make html` (Linux/Mac)
- `export PYTHONWARNINGS=` e si esegue `make html` (Linux/Mac)

¹⁰⁴⁰ <https://no-color.org/>

- set PYTHONWARNINGS= e si esegue make html (Windows)
- si modifica il Makefile/make.bat e si imposta la variabile d'ambiente

Vedere anche

`sphinx-quickstart(1)`

12.2 Applicazioni Aggiuntive

sphinx-apidoc

Sinossi

```
sphinx-apidoc    [OPTIONS]    -o    <OUTPUT_PATH>    <MODULE_PATH>  
[EXCLUDE_PATTERN ...]
```

Descrizione

sphinx-apidoc è uno strumento per la generazione automatica di sorgenti Sphinx che, utilizzando l'estensione [autodoc](#), documenta un intero pacchetto nello stile di altri tool di documentazione automatica di API.

MODULE_PATH è il path di un pacchetto Python da documentare e *OUTPUT_PATH* è la directory in cui vengono posizionati i sorgenti generati. Tutti gli *EXCLUDE_PATTERN* forniti sono pattern [fnmatch-style](#)¹⁰⁴¹ di file e/o directory che verranno esclusi dalla generazione.

Avvertimento: `sphinx-apidoc` genera sorgenti che utilizzano `sphinx.ext.autodoc` per documentare tutti i moduli trovati. Se qualche modulo ha effetti collaterali durante l'importazione, questi verranno eseguiti da `autodoc` quando viene eseguito `sphinx-build`.

Se si documentano gli script (al contrario dei moduli della libreria), la loro routine principale dev'essere protetta da una condizione `if __name__ == '__main__':`.

¹⁰⁴¹ <https://docs.python.org/3/library/fnmatch.html>

Opzioni

-o <OUTPUT_PATH>

Directory in cui posizionare i file di output. Se non esiste, viene creato.

-q

Non emette nulla sull'output standard, scrivere solo avvisi ed errori nello "standard error".

-f, --force

Forza la sovrascrittura di tutti i file generati esistenti.

-l, --follow-links

Segue i link simbolici. Il default è False.

-n, --dry-run

Non creare o rimuovere alcun file.

-s <suffix>

Suffisso per i file sorgenti generati. Il default è `rst`.

-d <MAXDEPTH>

Profondità massima per la generazione del file del sommario. Il default è 4.

--tocfile

Filename per un file di sommario. Il default è `modules`.

-T, --no-toc

Non crea un file di sommario. Ignorato quando viene fornito `--full`.

--remove-old

Rimuove i file esistenti nella directory di output che non vengono più creati. Non compatibile con `--full`.

-F, --full

Genera un progetto Sphinx completo (`conf.py`, `Makefile` ecc.) utilizzando lo stesso meccanismo di **sphinx-quickstart**.

-e, --separate

Inserisce la documentazione per ciascun modulo nella propria pagina.

Added in version 1.2.

-E, --no-headings

Non crea intestazioni (titoli) per i moduli/pacchetti. Ciò è utile, ad esempio, quando le docstring contengono già intestazioni.

-P, --private

Include i moduli `"_private"`.

Added in version 1.2.

--implicit-namespaces

Per default sphinx-apidoc elabora `sys.path` cercando solo i moduli. Python 3.3 ha introdotto i namespace impliciti [PEP 420](#)¹⁰⁴² che consentono strutture di percorso dei moduli come `foo/bar/module.py` o `foo/bar/baz/__init__.py` (notare che `bar` e `foo` sono namespace, non moduli).

Interpreta i path in modo ricorsivo secondo PEP-0420.

-M, --module-first

Mette la documentazione del modulo prima di quella del sottomodulo.

Queste opzioni vengono utilizzate quando viene specificato **--full**:

-a

Aggiunge `module_path` a `sys.path`.

-H <project>

Imposta il nome del progetto da inserire nei file generati (vedere [project](#)).

-A <author>

Imposta il/i nome/i dell'autore da inserire nei file generati (vedere [copyright](#)).

-V <version>

Imposta la versione del progetto da inserire nei file generati (vedere [version](#)).

-R <release>

Imposta la release del progetto da inserire nei file generati (vedere [release](#)).

Template di progetto

Added in version 2.2: Opzioni per i template di progetto per sphinx-apidoc

-t, --templatedir=TEMPLATEDIR

La directory per i file template. È possibile modificare i template dei file di progetto sphinx generati da apidoc. Sono consentiti i seguenti file template Jinja2:

- `module.rst_t`
- `package.rst_t`
- `toc.rst_t`
- `root_doc.rst_t`
- `conf.py_t`
- `Makefile_t`
- `Makefile.new_t`
- `make.bat_t`
- `make.bat.new_t`

¹⁰⁴² <https://peps.python.org/pep-0420/>

Nel dettaglio, fare riferimento ai file template di sistema forniti da Sphinx. (`sphinx/templates/apidoc` e `sphinx/templates/quickstart`)

Environment

SPHINX_APIDOC_OPTIONS

Un elenco di opzioni separate da virgole da aggiungere alle direttive automodule generate. Il default è `members,undoc-members,show-inheritance`.

Vedere anche

sphinx-build(1), *sphinx-autogen(1)*

sphinx-autogen

Sinossi

sphinx-autogen [*options*] <sourcefile> ...

Descrizione

sphinx-autogen è uno strumento per la generazione automatica di sorgenti Sphinx che, utilizzando l'estensione *autodoc*, documenta gli elementi inclusi nella lista *autosummary*.

sourcefile è il percorso di uno o più documenti reStructuredText contenenti voci *autosummary* con l'opzione `:toctree::` settata. *sourcefile* può essere un pattern in stile *fnmatch*¹⁰⁴³.

Opzioni

-o <outputdir>

Directory in cui posizionare il file di output. Se non esiste, viene creato. Il valore di default quello passato all'opzione `:toctree::`.

-s <suffix>, **--suffix** <suffix>

Il suffisso di default da utilizzare per i file generati. Il default è `rst`.

-t <templates>, **--templates** <templates>

Directory dei template personalizzati. Il default è `None`.

-i, **--imported-members**

Documenta i membri importati.

-a, **--respect-module-all**

Documenta esattamente i membri nell'attributo `__all__` di un modulo.

¹⁰⁴³ <https://docs.python.org/3/library/fnmatch.html#module-fnmatch>

--remove-old

Rimuove i file esistenti nella directory di output che non vengono più generati.

Esempio

Data la seguente struttura di directory:

```
docs
├── index.rst
├── ...
foobar
├── foo
│   └── __init__.py
├── bar
│   ├── __init__.py
│   └── baz
│       └── __init__.py
```

e presupponendo che docs/index.rst contenga quanto segue:

```
Modules
=====

.. autosummary::
   :toctree: modules

   foobar.foo
   foobar.bar
   foobar.bar.baz
```

Se si esegue quanto segue:

```
$ PYTHONPATH=. sphinx-autogen docs/index.rst
```

poi verranno creati i seguenti file stub in docs:

```
docs
├── index.rst
├── modules
│   ├── foobar.bar.rst
│   ├── foobar.bar.baz.rst
│   └── foobar.foo.rst
```

e ciascuno di questi file conterrà una direttiva `autodoc` e alcune altre informazioni.

Vedere anche

sphinx-build(1), *sphinx-apidoc(1)*

builder

Una classe (che eredita da *Builder*) che prende documenti analizzati ed esegue un'azione su di essi. Normalmente, i builder traducono i documenti in un formato di output, ma è anche possibile utilizzarli ad es. per verificare la presenza di link interrotti nella documentazione o creare informazioni sulla copertura.

Consultare *I Builder* per una panoramica sui builder nativi di Sphinx.

directory di configurazione

La directory contenente `conf.py`. Per default, è uguale alla *source directory*, ma può essere impostato diversamente con l'opzione della riga di comando `-c`.

direttiva

Un elemento di markup reStructuredText che consente di contrassegnare un blocco di contenuto con un significato speciale. Le direttive vengono fornite non solo dai docutil, ma Sphinx e le estensioni personalizzate possono aggiungerne di proprie. La sintassi della direttiva di base è simile alla seguente:

```
.. directivename:: argument ...  
   :option: value  
  
   Content of the directive.
```

Consultare *Direttive* per ulteriori informazioni.

document name

Poiché i file sorgenti reST possono avere estensioni diverse (ad alcuni piace `.txt`, ad altri `.rst` -- l'estensione può essere configurata con *source_suffix*) e diversi sistemi operativi hanno separatori di path diversi, Sphinx li astrae: *document names* sono sempre relativi alla *source directory*, l'estensione viene eliminata e i separatori di path vengono

convertiti in slash. Tutti i valori, parametri e simili che si riferiscono a "documents" si aspettano tali nomi di documento.

Esempi di nomi di documenti sono `index`, `library/zipfile` o `reference/datamodel/types`. Notare che non è presente alcuna slash iniziale o finale.

dominio

Un dominio è una raccolta di markup (*directive* e *role* reStructuredText) per descrivere e creare link a relativi *object*, ad es. elementi di un linguaggio di programmazione. I nomi delle direttive e dei ruoli in un dominio hanno nomi come `domain:name`, ad es. `py:function`.

Avere domini significa che non ci sono problemi di nomenclatura quando si vuole fare riferimento a una serie di documentazione, ad es. classi C++ e Python. Significa anche che le estensioni che supportano la documentazione di linguaggi completamente nuovi sono molto più facili da scrivere.

Per ulteriori informazioni, fare riferimento a *Domini*.

environment

Una struttura in cui le informazioni su tutti i documenti sotto la radice vengono salvate e utilizzate per i riferimenti incrociati. L'environment [*ambiente*] viene [pickled] dopo la fase di analisi, in modo che le esecuzioni successive debbano solo leggere e analizzare i documenti nuovi e modificati.

estensione

Un *role*, *directive* personalizzato o un altro aspetto di Sphinx che consente agli utenti di modificare qualsiasi aspetto del processo di build all'interno di Sphinx.

Per ulteriori informazioni, fare riferimento a *Estensioni*.

documento master

Il documento che contiene la direttiva root *toctree*.

documento root

Lo stesso di *master document*.

oggetto

L'elemento base della documentazione di Sphinx. Ogni "direttiva object" (ad esempio *py:function* o *object*) crea un tale blocco; e alla maggior parte degli oggetti è possibile fare riferimenti incrociati.

RemoveInSphinxXXXWarning

Warning sulla funzionalità che verrà rimossa nella versione Sphinx-XXX. Di solito è causato dalle estensioni Sphinx il cui utilizzo è deprecato. Vedere anche *Deprecazione dei Warning*.

ruolo

Un elemento di markup reStructuredText che consente di contrassegnare una porzione di testo. Come le direttive, i ruoli sono estensibili. La sintassi di base è simile a questa: `:rolename: `content``. Vedere *Markup in linea* per i dettagli.

directory sorgente

La directory che, incluse le sue sottodirectory, contiene tutti i file sorgenti per un progetto Sphinx.

reStructuredText

Una sintassi di markup e un sistema di parser di testo in chiaro di facile lettura, what-you-see-is-what-you-get ("ciò che vedi è ciò che ottieni").

14.1 Release 7.4.0 (in sviluppo)

Dipendenze

Modifiche incompatibili

Deprecato

Funzionalità aggiunte

- [#11165](#)¹⁰⁴⁴: Support the [officially recommended](#)¹⁰⁴⁵ `.jinja` suffix for template files. Patch by James Addison and Adam Turner
- Flatten `Union[Literal[T], Literal[U], ...]` to `Literal[T, U, ...]` when turning annotations into strings. Patch by Adam Turner.
- [#12319](#)¹⁰⁴⁶: `sphinx.ext.extlinks`: Add `extlink-{name}` CSS class to links. Patch by Hugo van Kemenade.
- [#12387](#)¹⁰⁴⁷: Improve CLI progress message, when copying assets. Patch by INADA Nakoi and Bénédict Tran.
- [#12361](#)¹⁰⁴⁸: Add `BuildEnvironment.parser`. Patch by Chris Sewell.

¹⁰⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/11165>

¹⁰⁴⁵ <https://jinja.palletsprojects.com/en/latest/templates/#template-file-extension>

¹⁰⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/12319>

¹⁰⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/12387>

¹⁰⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/12361>

- [#12358](#)¹⁰⁴⁹: Add `Sphinx.fresh_env_used`. Patch by Chris Sewell.
- [#12329](#)¹⁰⁵⁰: Add detection of ambiguous `std:label` and `std:term` references during loading and resolution of Intersphinx targets. Patch by James Addison.
- [#12422](#)¹⁰⁵¹: Do not duplicate "navigation" in aria-label of built-in themes. Patch by Thomas Weißschuh
- [#12421](#)¹⁰⁵²: Include project name in `logo_alt` of built-in themes. Patch by Thomas Weißschuh
- [#12448](#)¹⁰⁵³: Add `sphinx-apidoc --remove-old` option. Patch by Chris Sewell.
- [#12456](#)¹⁰⁵⁴: Add `sphinx-autogen --remove-old` option. Patch by Chris Sewell.
- [#12479](#)¹⁰⁵⁵: Add warning subtype `toc.no_title`. Patch by Ondřej Navrátil.
- [#12492](#)¹⁰⁵⁶: Add helper methods for parsing reStructuredText content into nodes from within a directive.
 - `parse_content_to_nodes()` parses the directive's content and returns a list of Docutils nodes.
 - `parse_text_to_nodes()` parses the provided text and returns a list of Docutils nodes.
 - `parse_inline()` parses the provided text into inline elements and text nodes.Patch by Adam Turner.
- [#12258](#)¹⁰⁵⁷: Support `typing_extensions.Unpack` Patch by Bénédict Tran and Adam Turner.

Bugs fixed

- [#12314](#)¹⁰⁵⁸: Properly format `collections.abc.Callable` in annotations. Patch by Adam Turner.
- [#12162](#)¹⁰⁵⁹: Fix a performance regression in the C domain that has been present since version 3.0.0. Patch by Donald Hunter.
- [#12320](#)¹⁰⁶⁰: Fix removal of anchors from search summaries (regression in 7.3.0). Patch by Will Lachance.

¹⁰⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/12358>

¹⁰⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/12329>

¹⁰⁵¹ <https://github.com/sphinx-doc/sphinx/issues/12422>

¹⁰⁵² <https://github.com/sphinx-doc/sphinx/issues/12421>

¹⁰⁵³ <https://github.com/sphinx-doc/sphinx/issues/12448>

¹⁰⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/12456>

¹⁰⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/12479>

¹⁰⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/12492>

¹⁰⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/12258>

¹⁰⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/12314>

¹⁰⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/12162>

¹⁰⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/12320>

- [#12251](#)¹⁰⁶¹: Fix `merge_domaindata()` in `sphinx.ext.duration`. Patch by Matthias Geier.
- [#12224](#)¹⁰⁶²: Properly detect WebP files. Patch by Benjamin Cabé.
- [#12380](#)¹⁰⁶³: LaTeX: Footnote mark sometimes indicates Page N where N is the current page number and the footnote does appear on that same page. Patch by Jean-François B.
- [#12416](#)¹⁰⁶⁴: `root_doc` is synchronized with `master_doc` so that if either of the two values is modified, the other reflects that modification. It is still recommended to use `root_doc`. Patch by Bénédict Tran.
- [#12220](#)¹⁰⁶⁵: Fix loading custom template translations for en locale. Patch by Nicolas Peugnet.
- [#12459](#)¹⁰⁶⁶: Add valid-type arguments to the `linkcheck_rate_limit_timeout` configuration setting. Patch by James Addison.
- [#12331](#)¹⁰⁶⁷: Resolve data-URI-image-extraction regression from v7.3.0 affecting builders without native support for data-URIs in their output format. Patch by James Addison.
- [#12494](#)¹⁰⁶⁸: Fix invalid `genindex.html` file produced with translated docs (regression in 7.1.0). Patch by Nicolas Peugnet.

Testing

- karma: refactor HTML search tests to use fixtures generated by Sphinx. Patch by James Addison.

14.2 Release 7.3.7 (released Apr 19, 2024)

Bugs fixed

- [#12299](#)¹⁰⁶⁹: Defer loading themes defined via entry points until their explicit use by the user or a child theme. Patch by Adam Turner.
- [#12305](#)¹⁰⁷⁰: Return the default value for `theme.get_config()` with an unsupported theme configuration section. Patch by Adam Turner.

¹⁰⁶¹ <https://github.com/sphinx-doc/sphinx/issues/12251>

¹⁰⁶² <https://github.com/sphinx-doc/sphinx/issues/12224>

¹⁰⁶³ <https://github.com/sphinx-doc/sphinx/issues/12380>

¹⁰⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/12416>

¹⁰⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/12220>

¹⁰⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/12459>

¹⁰⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/12331>

¹⁰⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/12494>

¹⁰⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/12299>

¹⁰⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/12305>

14.3 Release 7.3.6 (released Apr 17, 2024)

Bugs fixed

- [#12295](#)¹⁰⁷¹: Re-export all AST types in the C and C++ domains. Patch by Adam Turner.
- [#12295](#)¹⁰⁷²: Re-export various objects from `sphinx.domains.python._annotations` in `sphinx.domains.python`. Patch by Jacob Chesslo and Adam Turner.

14.4 Release 7.3.5 (released Apr 17, 2024)

Bugs fixed

- [#12295](#)¹⁰⁷³: Re-export various objects from `sphinx.domains.python._object` in `sphinx.domains.python`. Patch by Jacob Chesslo and Adam Turner.

14.5 Release 7.3.4 (released Apr 17, 2024)

Bugs fixed

- Handle cases when `Any` is not an instance of `type`. Patch by Adam Turner.

14.6 Release 7.3.3 (released Apr 17, 2024)

Bugs fixed

- [#12290](#)¹⁰⁷⁴: Fix a false-positive warning when setting a configuration value with `Any` as the valid type to a type other than the value's default. Patch by Adam Turner.

¹⁰⁷¹ <https://github.com/sphinx-doc/sphinx/issues/12295>

¹⁰⁷² <https://github.com/sphinx-doc/sphinx/issues/12295>

¹⁰⁷³ <https://github.com/sphinx-doc/sphinx/issues/12295>

¹⁰⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/12290>

14.7 Release 7.3.2 (released Apr 17, 2024)

Bugs fixed

- Preload all themes defined via entry points. Patch by Adam Turner.
- Fix a bad interaction between the 'Furo' theme and the new-style for configuration values. Patch by Adam Turner.

14.8 Release 7.3.1 (released Apr 17, 2024)

Dipendenze

- Require `tomli` on Python 3.10 and earlier. Patch by Adam Turner.

14.9 Release 7.3.0 (released Apr 16, 2024)

Dipendenze

- [#11858](#)¹⁰⁷⁵: Increase the minimum supported version of Alabaster to 0.7.14. Patch by Adam Turner.
- [#11411](#)¹⁰⁷⁶: Support Docutils 0.21¹⁰⁷⁷. Patch by Adam Turner.
- [#12012](#)¹⁰⁷⁸: Use `types-docutils` instead of `docutils-stubs`.

Deprecato

- [#11693](#)¹⁰⁷⁹: Support for old-style Makefile and `make.bat` output in **sphinx-quickstart**, and the associated options `-M`, `-m`, `--no-use-make-mode`, and `--use-make-mode`.
- [#11285](#)¹⁰⁸⁰: Direct access to `sphinx.testing.util.SphinxTestApp._status` or `sphinx.testing.util.SphinxTestApp._warning` is deprecated. Use the public properties `sphinx.testing.util.SphinxTestApp.status` and `sphinx.testing.util.SphinxTestApp.warning` instead. Patch by B  n  dikt Tran.
- tests: `sphinx.testing.util.strip_escseq()` is deprecated in favour of `sphinx.util.console.strip_colors()`. Patch by B  n  dikt Tran.

¹⁰⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/11858>

¹⁰⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/11411>

¹⁰⁷⁷ <https://docutils.sourceforge.io/RELEASE-NOTES.html#release-0-21-2024-04-09>

¹⁰⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/12012>

¹⁰⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/11693>

¹⁰⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/11285>

Funzionalità aggiunte

- [#12265](#)¹⁰⁸¹: Support theme configuration via `theme.toml`.
- [#11701](#)¹⁰⁸²: HTML Search: Adopt the new `<search>`¹⁰⁸³ element. Patch by Bénédict Tran.
- [#11776](#)¹⁰⁸⁴: Add long option names to `sphinx-build`. Patch by Hugo van Kemenade, Adam Turner, Bénédict Tran, and Ezio Melotti.
- Organise the `sphinx-build` options into groups. Patch by Adam Turner.
- [#11855](#)¹⁰⁸⁵: Defer computation of configuration values. Patch by Adam Turner.
- Add `:no-search:` as an alias of the `:nosearch:` metadata field. Patch by Adam Turner.
- [#11803](#)¹⁰⁸⁶: autodoc: Use an overridden `__repr__()` function in an enum, if defined. Patch by Shengyu Zhang.
- [#11825](#)¹⁰⁸⁷: Allow custom targets in the manpage role. Patch by Nicolas Peugnet.
- [#11892](#)¹⁰⁸⁸: Improved performance when resolving cross references in the C++ domain. Patch by Rouslan Korneychuk.
- [#11905](#)¹⁰⁸⁹: Add a `versionremoved` directive. Patch by Hugo van Kemenade, Adam Turner, and C.A.M. Gerlach.
- [#11981](#)¹⁰⁹⁰: Improve rendering of signatures using slice syntax, e.g., `def foo(arg: np.float64[:, :]) -> None:`
- The manpage builder now adds `OSC 8`¹⁰⁹¹ anchors to hyperlinks, using the `groff`¹⁰⁹² device control command.
- [#11015](#)¹⁰⁹³: Change the text of the `versionadded` directive from `New in [...]` to `Added in [...]`. Patch by Bénédict Tran.
- [#12131](#)¹⁰⁹⁴: Added `show_warning_types` configuration option. Patch by Chris Sewell.
- [#12193](#)¹⁰⁹⁵: Improve `external` warnings for unknown roles. In particular, suggest related role names if an object type is mistakenly used. Patch by Chris Sewell.

¹⁰⁸¹ <https://github.com/sphinx-doc/sphinx/issues/12265>

¹⁰⁸² <https://github.com/sphinx-doc/sphinx/issues/11701>

¹⁰⁸³ <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/search>

¹⁰⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/11776>

¹⁰⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/11855>

¹⁰⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/11803>

¹⁰⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/11825>

¹⁰⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/11892>

¹⁰⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/11905>

¹⁰⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/11981>

¹⁰⁹¹ <https://gist.github.com/egmontkob/eb114294efbcd5adb1944c9f3cb5feda>

¹⁰⁹² <https://lists.gnu.org/archive/html/groff/2021-10/msg00000.html>

¹⁰⁹³ <https://github.com/sphinx-doc/sphinx/issues/11015>

¹⁰⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/12131>

¹⁰⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/12193>

- Add public type alias `sphinx.util.typing.ExtensionMetadata`. This can be used by extension developers to annotate the return type of their `setup` function. Patch by Chris Sewell.

Bugs fixed

- [#11668](#)¹⁰⁹⁶: Raise a useful error when `theme.conf` is missing. Patch by Vinay Sajip.
- [#11622](#)¹⁰⁹⁷: Ensure that the order of keys in `searchindex.js` is deterministic. Patch by Pietro Albini.
- [#11617](#)¹⁰⁹⁸: ANSI control sequences are stripped from the output when writing to a warnings file with `-W`. Patch by Bénédict Tran.
- [#11666](#)¹⁰⁹⁹: Skip all hidden directories in `CatalogRepository.pofiles`. Patch by Aryaz Eghbali.
- [#9686](#)¹¹⁰⁰: html builder: Fix MathJax lazy loading when equations appear in titles. Patch by Bénédict Tran.
- [#11483](#)¹¹⁰¹: singlehtml builder: Fix MathJax lazy loading when the index does not contain any math equations. Patch by Bénédict Tran.
- [#11697](#)¹¹⁰²: HTML Search: add 'noindex' meta robots tag. Patch by James Addison.
- [#11678](#)¹¹⁰³: Fix a possible `ZeroDivisionError` in `sphinx.ext.coverage`. Patch by Stephen Finucane.
- [#11756](#)¹¹⁰⁴: LaTeX: build error with recent TeXLive due to missing `substitutefont` package (triggered if using `fontenc` with T2A option and document language is not a Cyrillic one). Patch by Jean-François B.
- [#11675](#)¹¹⁰⁵: Fix rendering of progression bars in environments that do not support ANSI control sequences. Patch by Bénédict Tran.
- [#11861](#)¹¹⁰⁶: Whitelist more types with an incorrect `__module__` attribute. Patch by Adam Turner.
- [#11715](#)¹¹⁰⁷: Apply `tls_verify` and `tls_cacerts` config to `ImageDownloader`. Patch by Nick Touran.

¹⁰⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/11668>

¹⁰⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/11622>

¹⁰⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/11617>

¹⁰⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/11666>

¹¹⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/9686>

¹¹⁰¹ <https://github.com/sphinx-doc/sphinx/issues/11483>

¹¹⁰² <https://github.com/sphinx-doc/sphinx/issues/11697>

¹¹⁰³ <https://github.com/sphinx-doc/sphinx/issues/11678>

¹¹⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/11756>

¹¹⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/11675>

¹¹⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/11861>

¹¹⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/11715>

- Allow hyphens in group names for *productionlist* cross-references. Patch by Adam Turner.
- #11433¹¹⁰⁸: Added the *linkcheck_allow_unauthorized* configuration option. Set this option to False to report HTTP 401 (unauthorized) server responses as broken. Patch by James Addison.
- #11868¹¹⁰⁹: linkcheck: added a distinct timeout reporting status code. This can be enabled by setting *linkcheck_report_timeouts_as_broken* to False. Patch by James Addison.
- #11869¹¹¹⁰: Refresh the documentation for the *linkcheck_timeout* setting. Patch by James Addison.
- #11874¹¹¹¹: Configure a default 30-second value for *linkcheck_timeout*. Patch by James Addison.
- #11886¹¹¹²: Print the Jinja2 template path chain in *TemplateNotFound* exceptions. Patch by Colin Marquardt.
- #11598¹¹¹³: Do not use query components in URLs for assets in EPUB rendering. Patch by David Runge.
- #11904¹¹¹⁴: Support unary subtraction when parsing annotations. Patch by James Addison.
- #11925¹¹¹⁵: Blacklist the *sphinxprettysearchresults* extension; the functionality it provides was merged into Sphinx v2.0.0. Patch by James Addison.
- #11917¹¹¹⁶: Fix rendering of annotated inherited members for Python 3.9. Patch by Janet Carson.
- #11935¹¹¹⁷: C Domain: Fix namespace-pop context. Patch by Frank Dana.
- #11923¹¹¹⁸: Avoid zombie processes when parallel builds fail. Patch by Felix von Drigalski.
- #11353¹¹¹⁹: Support enumeration classes inheriting from mixin or data types. Patch by Bénédict Tran.
- #11962¹¹²⁰: Fix target resolution when using *:paramtype:* fields. Patch by Bénédict Tran.

¹¹⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/11433>

¹¹⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/11868>

¹¹¹⁰ <https://github.com/sphinx-doc/sphinx/issues/11869>

¹¹¹¹ <https://github.com/sphinx-doc/sphinx/issues/11874>

¹¹¹² <https://github.com/sphinx-doc/sphinx/issues/11886>

¹¹¹³ <https://github.com/sphinx-doc/sphinx/issues/11598>

¹¹¹⁴ <https://github.com/sphinx-doc/sphinx/issues/11904>

¹¹¹⁵ <https://github.com/sphinx-doc/sphinx/issues/11925>

¹¹¹⁶ <https://github.com/sphinx-doc/sphinx/issues/11917>

¹¹¹⁷ <https://github.com/sphinx-doc/sphinx/issues/11935>

¹¹¹⁸ <https://github.com/sphinx-doc/sphinx/issues/11923>

¹¹¹⁹ <https://github.com/sphinx-doc/sphinx/issues/11353>

¹¹²⁰ <https://github.com/sphinx-doc/sphinx/issues/11962>

- [#11944](#)¹¹²¹: Use anchor in search preview. Patch by Will Lachance.
- [#12008](#)¹¹²²: Fix case-sensitive lookup of `std:label` names in intersphinx inventory. Patch by Michael Goerz.
- [#11958](#)¹¹²³: HTML Search: Fix partial matches overwriting full matches. Patch by William Lachance.
- [#11959](#)¹¹²⁴: Fix multiple term matching when word appears in both title and document. Patch by Will Lachance.
- [#11474](#)¹¹²⁵: Fix doctrees caching causing files not be rebuilt in some cases, e.g., when `numfig` is True. Patch by Bénédict Tran.
- [#11278](#)¹¹²⁶: autodoc: Fix rendering of `functools.singledispatchmethod`¹¹²⁷ combined with `@classmethod`¹¹²⁸. Patch by Bénédict Tran.
- [#11894](#)¹¹²⁹: Do not add checksums to css files if building using the htmlhelp builder. Patch by reduerK akiM.
- [#12052](#)¹¹³⁰: Remove `<script>` and `<style>` tags from the content of search result summary snippets. Patch by James Addison.
- [#11578](#)¹¹³¹: HTML Search: Order non-main index entries after other results. Patch by Brad King.
- [#12147](#)¹¹³²: autosummary: Fix a bug whereby the wrong file extension may be used, when multiple suffixes are specified in `source_suffix`. Patch by Sutou Kouhei.
- [#10786](#)¹¹³³: improve the error message when a file to be copied (e.g., an asset) is removed during Sphinx execution. Patch by Bénédict Tran.
- [#12040](#)¹¹³⁴: HTML Search: Ensure that document titles that are partially-matched by the user search query are included in search results. Patch by James Addison.
- [#11970](#)¹¹³⁵: singlehtml builder: make target URIs to be same-document references in the sense of [RFC 3986, §4.4](#)¹¹³⁶, e.g., `index.html#foo` becomes `#foo`. Patch by Eric Norige.

¹¹²¹ <https://github.com/sphinx-doc/sphinx/issues/11944>

¹¹²² <https://github.com/sphinx-doc/sphinx/issues/12008>

¹¹²³ <https://github.com/sphinx-doc/sphinx/issues/11958>

¹¹²⁴ <https://github.com/sphinx-doc/sphinx/issues/11959>

¹¹²⁵ <https://github.com/sphinx-doc/sphinx/issues/11474>

¹¹²⁶ <https://github.com/sphinx-doc/sphinx/issues/11278>

¹¹²⁷ <https://docs.python.org/3/library/functools.html#functools.singledispatchmethod>

¹¹²⁸ <https://docs.python.org/3/library/functions.html#classmethod>

¹¹²⁹ <https://github.com/sphinx-doc/sphinx/issues/11894>

¹¹³⁰ <https://github.com/sphinx-doc/sphinx/issues/12052>

¹¹³¹ <https://github.com/sphinx-doc/sphinx/issues/11578>

¹¹³² <https://github.com/sphinx-doc/sphinx/issues/12147>

¹¹³³ <https://github.com/sphinx-doc/sphinx/issues/10786>

¹¹³⁴ <https://github.com/sphinx-doc/sphinx/issues/12040>

¹¹³⁵ <https://github.com/sphinx-doc/sphinx/issues/11970>

¹¹³⁶ <https://datatracker.ietf.org/doc/html/rfc3986.html#section-4.4>

- [#12271](#)¹¹³⁷: Partially revert Docutils' [r9562](#)¹¹³⁸ to fix EPUB files. Patch by Adam Turner.
- [#12253](#)¹¹³⁹: Escape reserved path characters in the remote images post-transform download cache. Patch by James Addison and Adam Turner.

Testing

- Reorganise tests into directories. Patch by Adam Turner.
- Clean up global state in `SphinxTestApp`. Patch by Adam Turner.
- [#11285](#)¹¹⁴⁰: `pytest.mark.sphinx()` and `sphinx.testing.util.SphinxTestApp` accept *warningiserror*, *keep_going* and *verbosity* as keyword arguments. Patch by B  n  dikt Tran.
- [#11285](#)¹¹⁴¹: `sphinx.testing.util.SphinxTestApp` *status* and *warning* arguments are checked to be `io.StringIO`¹¹⁴² objects (the public API incorrectly assumed this without checking it). Patch by B  n  dikt Tran.
- Report the result of `test_run_epubcheck` as `skipped` instead of `success` when either Java or `epubcheck` are not available.
- Use dynamic allocation of unused port numbers for the test HTTP(S) servers. As a side-effect, this removes the need for test server lockfiles, meaning that any remaining `tests/test-server.lock` files can safely be deleted.

14.10 Release 7.2.6 (released Sep 13, 2023)

Bugs fixed

- [#11679](#)¹¹⁴³: Add the `SPHINX_AUTODOC_RELOAD_MODULES` environment variable, which if set reloads modules when using autodoc with `TYPE_CHECKING = True`. Patch by Matt Wozniski and Adam Turner.
- [#11679](#)¹¹⁴⁴: Use `importlib.reload()`¹¹⁴⁵ to reload modules in autodoc. Patch by Matt Wozniski and Adam Turner.

¹¹³⁷ <https://github.com/sphinx-doc/sphinx/issues/12271>

¹¹³⁸ <https://sourceforge.net/p/docutils/code/9562/>

¹¹³⁹ <https://github.com/sphinx-doc/sphinx/issues/12253>

¹¹⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/11285>

¹¹⁴¹ <https://github.com/sphinx-doc/sphinx/issues/11285>

¹¹⁴² <https://docs.python.org/3/library/io.html#io.StringIO>

¹¹⁴³ <https://github.com/sphinx-doc/sphinx/issues/11679>

¹¹⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/11679>

¹¹⁴⁵ <https://docs.python.org/3/library/importlib.html#importlib.reload>

14.11 Release 7.2.5 (released Aug 30, 2023)

Bugs fixed

- [#11645](#)¹¹⁴⁶: Fix a regression preventing autodoc from importing modules within packages that make use of `if typing.TYPE_CHECKING:` to guard circular imports needed by type checkers. Patch by Matt Wozniski.
- [#11634](#)¹¹⁴⁷: Fixed inheritance diagram relative link resolution for sibling files in a subdirectory. Patch by Albert Shih.
- [#11659](#)¹¹⁴⁸: Allow `?config=...` in `mathjax_path`.
- [#11654](#)¹¹⁴⁹: autodoc: Fail with a more descriptive error message when an object claims to be an instance of `type`, but is not a class. Patch by James Braza.
- 11620: Cease emitting `source-read` events for files read via the `include`¹¹⁵⁰ directive.
- 11620: Add a new `include-read` for observing and transforming the content of included files via the `include`¹¹⁵¹ directive.
- [#11627](#)¹¹⁵²: Restore support for copyright lines of the form `YYYY` when `SOURCE_DATE_EPOCH` is set.

14.12 Release 7.2.4 (released Aug 28, 2023)

Bugs fixed

- [#11618](#)¹¹⁵³: Fix a regression in the `MoveModuleTargets` transform, introduced in [#10478](#)¹¹⁵⁴ ([#9662](#)¹¹⁵⁵).
- [#11649](#)¹¹⁵⁶: linkcheck: Resolve hanging tests for timezones west of London and incorrect conversion from UTC to offsets from the UNIX epoch. Patch by Dmitry Shachnev and Adam Turner.

¹¹⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/11645>

¹¹⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/11634>

¹¹⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/11659>

¹¹⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/11654>

¹¹⁵⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

¹¹⁵¹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

¹¹⁵² <https://github.com/sphinx-doc/sphinx/issues/11627>

¹¹⁵³ <https://github.com/sphinx-doc/sphinx/issues/11618>

¹¹⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/10478>

¹¹⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/9662>

¹¹⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/11649>

14.13 Release 7.2.3 (released Aug 23, 2023)

Dipendenze

- [#11576](#)¹¹⁵⁷: Require sphinxcontrib-serializinghtml 1.1.9.

Bugs fixed

- Fix regression in `autodoc.Documenter.parse_name()`.
- Fix regression in JSON serialisation.
- [#11543](#)¹¹⁵⁸: autodoc: Support positional-only parameters in classmethod methods when `autodoc_preserve_defaults` is `True`.
- Restore support string methods on path objects. This is deprecated and will be removed in Sphinx 8. Use `os.fspath()`¹¹⁵⁹ to convert `Path`¹¹⁶⁰ objects to strings, or `Path`¹¹⁶¹'s methods to work with path objects.

14.14 Release 7.2.2 (released Aug 17, 2023)

Bugs fixed

- Fix the signature of the `StateMachine.insert_input()` patch, for when calling with keyword arguments.
- Fixed membership testing (`in`) for the `str`¹¹⁶² interface of the asset classes (`_CascadingStyleSheet` and `_JavaScript`), which several extensions relied upon.
- Fixed a type error in `SingleFileHTMLBuilder._get_local_toc tree`, `includehidden` may be passed as a string or a boolean.
- Fix `:noindex:` for `PyModule` and `JSModule`.

¹¹⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/11576>

¹¹⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/11543>

¹¹⁵⁹ <https://docs.python.org/3/library/os.html#os.fspath>

¹¹⁶⁰ <https://docs.python.org/3/library/pathlib.html#pathlib.Path>

¹¹⁶¹ <https://docs.python.org/3/library/pathlib.html#pathlib.Path>

¹¹⁶² <https://docs.python.org/3/library/stdtypes.html#str>

14.15 Release 7.2.1 (released Aug 17, 2023)

Bugs fixed

- Restored the the `str`¹¹⁶³ interface of the asset classes (`_CascadingStyleSheet` and `_JavaScript`), which several extensions relied upon. This will be removed in Sphinx 9.
- Restored calls to `Builder.add_{css,js}_file()`, which several extensions relied upon.
- Restored the private API `TocTree.get_toctree_ancestors()`, which several extensions relied upon.

14.16 Release 7.2.0 (released Aug 17, 2023)

Dipendenze

- [#11511](#)¹¹⁶⁴: Drop Python 3.8 support.
- [#11576](#)¹¹⁶⁵: Require Pygments 2.14 or later.

Deprecato

- [#11512](#)¹¹⁶⁶: Deprecate `sphinx.util.md5` and `sphinx.util.sha1`. Use `hashlib` instead.
- [#11526](#)¹¹⁶⁷: Deprecate `sphinx.testing.path`. Use `os.path` or `pathlib` instead.
- [#11528](#)¹¹⁶⁸: Deprecate `sphinx.util.split_index_msg` and `sphinx.util.split_into`. Use `sphinx.util.index_entries.split_index_msg` instead.
- Deprecate `sphinx.builders.html.Stylesheet` and `sphinx.builders.html.Javascript`. Use `sphinx.application.Sphinx.add_css_file()` and `sphinx.application.Sphinx.add_js_file()` instead.
- [#11582](#)¹¹⁶⁹: Deprecate `sphinx.builders.html.StandaloneHTMLBuilder.css_files` and `sphinx.builders.html.StandaloneHTMLBuilder.script_files`. Use `sphinx.application.Sphinx.add_css_file()` and `sphinx.application.Sphinx.add_js_file()` instead.

¹¹⁶³ <https://docs.python.org/3/library/stdtypes.html#str>

¹¹⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/11511>

¹¹⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/11576>

¹¹⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/11512>

¹¹⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/11526>

¹¹⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/11528>

¹¹⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/11582>

- [#11459](#)¹¹⁷⁰: Deprecate `sphinx.ext.autodoc.preserve_defaults.get_function_def()`. Patch by Bénédict Tran.

Funzionalità aggiunte

- [#11526](#)¹¹⁷¹: Support `os.PathLike` types and `pathlib.Path` objects in many more places.
- [#5474](#)¹¹⁷²: coverage: Print summary statistics tables. Patch by Jorge Leitao.
- [#6319](#)¹¹⁷³: viewcode: Add `viewcode_line_numbers` to control whether line numbers are added to rendered source code. Patch by Ben Krikler.
- [#9662](#)¹¹⁷⁴: Add the `:no-typesetting:` option to suppress textual output and only create a linkable anchor. Patch by Latosha Maltba.
- [#11221](#)¹¹⁷⁵: C++: Support domain objects in the table of contents. Patch by Rouslan Korneychuk.
- [#10938](#)¹¹⁷⁶: doctest: Add `doctest_show_successes` option. Patch by Trey Hunner.
- [#11533](#)¹¹⁷⁷: Add `:no-index:`, `:no-index-entry:`, and `:no-contents-entry:`.
- [#11572](#)¹¹⁷⁸: Improve debug logging of reasons why files are detected as out of date. Patch by Eric Larson.
- [#10678](#)¹¹⁷⁹: Emit `source-read` events for files read via the `include`¹¹⁸⁰ directive. Patch by Halldor Fannar.
- [#11570](#)¹¹⁸¹: Use short names when using **PEP 585**¹¹⁸² built-in generics. Patch by Riccardo Mori.
- [#11300](#)¹¹⁸³: Improve `SigElementFallbackTransform` fallback logic and signature text elements nodes. See [the documentation](#) for more details. Patch by Bénédict Tran.
- Allow running Sphinx with `python -m sphinx build`

¹¹⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/11459>

¹¹⁷¹ <https://github.com/sphinx-doc/sphinx/issues/11526>

¹¹⁷² <https://github.com/sphinx-doc/sphinx/issues/5474>

¹¹⁷³ <https://github.com/sphinx-doc/sphinx/issues/6319>

¹¹⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/9662>

¹¹⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/11221>

¹¹⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/10938>

¹¹⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/11533>

¹¹⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/11572>

¹¹⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/10678>

¹¹⁸⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

¹¹⁸¹ <https://github.com/sphinx-doc/sphinx/issues/11570>

¹¹⁸² <https://peps.python.org/pep-0585/>

¹¹⁸³ <https://github.com/sphinx-doc/sphinx/issues/11300>

Bugs fixed

- [#11077](#)¹¹⁸⁴: graphviz: Fix relative links from within the graph. Patch by Ralf Grubenmann.
- [#11529](#)¹¹⁸⁵: Line Block in LaTeX builder outputs spurious empty token. Patch by Adrian Vollmer.
- [#11196](#)¹¹⁸⁶: autosummary: Summary line extraction failed with "e.g."
- [#10614](#)¹¹⁸⁷: Fixed a number of bugs in inheritance diagrams that resulted in missing or broken links. Patch by Albert Shih.
- [#9428](#)¹¹⁸⁸: Exclude substitution definitions when running the gettext builder. Patch by Alvin Wong.
- [#10795](#)¹¹⁸⁹: Raise a descriptive error if graphviz_dot is falsy.
- [#11546](#)¹¹⁹⁰: Translated nodes identical to their original text are now marked with the translated=True attribute.
- [#10049](#)¹¹⁹¹: html: Change "Permalink" to "Link" for title text in link anchors.
- [#4225](#)¹¹⁹²: Relax Pygments parsing on lexing failures.
- [#11246](#)¹¹⁹³: Allow inline links in the first line of a docstring and one-line type comments `#: :meta ...:` when using `sphinx.ext.napoleon`. Patch by B  n  dikt Tran.
- [#10930](#)¹¹⁹⁴: Highlight all search terms on the search results page. Patch by Dmitry Shachnev.
- [#11473](#)¹¹⁹⁵: Type annotations containing `Literal`¹¹⁹⁶ enumeration values now render correctly. Patch by B  n  dikt Tran.
- [#11591](#)¹¹⁹⁷: Fix support for C coverage in `sphinx.ext.coverage` extension. Patch by Stephen Finucane.
- [#11594](#)¹¹⁹⁸: HTML Theme: Enhancements to horizontal scrolling on smaller devices in the agogo theme. Patch by Lukas Engelter.

¹¹⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/11077>

¹¹⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/11529>

¹¹⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/11196>

¹¹⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/10614>

¹¹⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/9428>

¹¹⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/10795>

¹¹⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/11546>

¹¹⁹¹ <https://github.com/sphinx-doc/sphinx/issues/10049>

¹¹⁹² <https://github.com/sphinx-doc/sphinx/issues/4225>

¹¹⁹³ <https://github.com/sphinx-doc/sphinx/issues/11246>

¹¹⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/10930>

¹¹⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/11473>

¹¹⁹⁶ <https://docs.python.org/3/library/typing.html#typing.Literal>

¹¹⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/11591>

¹¹⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/11594>

- [#11459](#)¹¹⁹⁹: Fix support for async and lambda functions in `sphinx.ext.autodoc.preserve_defaults`. Patch by B  n  dikt Tran.

Testing

- [#11577](#)¹²⁰⁰: pytest: Fail tests on "XPASS".
- [#11577](#)¹²⁰¹: pytest: Use "importlib" import mode.
- [#11577](#)¹²⁰²: pytest: Set PYTHONWARNINGS=error.
- [#11577](#)¹²⁰³: pytest: Set strict config and strict markers.

14.17 Release 7.1.2 (released Aug 02, 2023)

Bugs fixed

- [#11542](#)¹²⁰⁴: linkcheck: Properly respect `linkcheck_anchors` and do not spuriously report failures to validate anchors. Patch by James Addison.

14.18 Release 7.1.1 (released Jul 27, 2023)

Bugs fixed

- [#11514](#)¹²⁰⁵: Fix `SOURCE_DATE_EPOCH` in multi-line copyright footer. Patch by B  n  dikt Tran.

14.19 Release 7.1.0 (released Jul 24, 2023)

Modifiche incompatibili

- Releases are no longer signed, given the [change in PyPI policy](#)¹²⁰⁶.

¹¹⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/11459>

¹²⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/11577>

¹²⁰¹ <https://github.com/sphinx-doc/sphinx/issues/11577>

¹²⁰² <https://github.com/sphinx-doc/sphinx/issues/11577>

¹²⁰³ <https://github.com/sphinx-doc/sphinx/issues/11577>

¹²⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/11542>

¹²⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/11514>

¹²⁰⁶ <https://blog.pyPI.org/posts/2023-05-23-removing-gpg/>

Deprecato

- [#11412](#)¹²⁰⁷: Emit warnings on using a deprecated Python-specific index entry type (namely, `module`, `keyword`, `operator`, `object`, `exception`, `statement`, and `builtin`) in the `index` directive, and set the removal version to Sphinx 9. Patch by Adam Turner.

Funzionalità aggiunte

- [#11415](#)¹²⁰⁸: Add a checksum to JavaScript and CSS asset URIs included within generated HTML, using the CRC32 algorithm.
- `require_sphinx()` now allows the version requirement to be specified as (major, minor).
- [#11011](#)¹²⁰⁹: Allow configuring a line-length limit for object signatures, via `maximum_signature_line_length` and the domain-specific variants. If the length of the signature (in characters) is greater than the configured limit, each parameter in the signature will be split to its own logical line. This behaviour may also be controlled by options on object description directives, for example `py:function:single-line-parameter-list`. Patch by Thomas Louf, Adam Turner, and Jean-François B.
- [#10983](#)¹²¹⁰: Support for multiline copyright statements in the footer block. Patch by Stefanie Molin
- `sphinx.util.display.status_iterator` now clears the current line with ANSI control codes, rather than overprinting with space characters.
- [#11431](#)¹²¹¹: linkcheck: Treat SSL failures as broken links. Patch by James Addison.
- [#11157](#)¹²¹²: Keep the translated attribute on translated nodes.
- [#11451](#)¹²¹³: Improve the traceback displayed when using `sphinx-build -T` in parallel builds. Patch by Bénédict Tran
- [#11324](#)¹²¹⁴: linkcheck: Use session-based HTTP requests.
- [#11438](#)¹²¹⁵: Add support for the `py:class` and `py:function` directives for PEP 695 (generic classes and functions declarations) and PEP 696 (default type parameters). Multi-line support ([#11011](#)¹²¹⁶) is enabled for type parameters list and can be locally controlled on object description directives, e.g., `py:function:single-line-type-parameter-list`. Patch by Bénédict Tran.

¹²⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/11412>

¹²⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/11415>

¹²⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/11011>

¹²¹⁰ <https://github.com/sphinx-doc/sphinx/issues/10983>

¹²¹¹ <https://github.com/sphinx-doc/sphinx/issues/11431>

¹²¹² <https://github.com/sphinx-doc/sphinx/issues/11157>

¹²¹³ <https://github.com/sphinx-doc/sphinx/issues/11451>

¹²¹⁴ <https://github.com/sphinx-doc/sphinx/issues/11324>

¹²¹⁵ <https://github.com/sphinx-doc/sphinx/issues/11438>

¹²¹⁶ <https://github.com/sphinx-doc/sphinx/issues/11011>

Testing

- [#11345](#)¹²²⁴: Always delete `docutils.conf` in test directories when running `SphinxTestApp.cleanup()`.

14.20 Release 7.0.1 (released May 12, 2023)

Dipendenze

- [#11411](#)¹²²⁵: Support [Docutils 0.20](#)¹²²⁶. Patch by Adam Turner.

Bugs fixed

- [#11418](#)¹²²⁷: Clean up remaining references to `sphinx.setup_command` following the removal of support for `setuptools`. Patch by Willem Mulder.

14.21 Release 7.0.0 (released Apr 29, 2023)

Modifiche incompatibili

- [#11359](#)¹²²⁸: Remove long-deprecated aliases for `MecabSplitter` and `DefaultSplitter` in `sphinx.search.ja`.
- [#11360](#)¹²²⁹: Remove deprecated `make_old_id` functions in domain object description classes.
- [#11363](#)¹²³⁰: Remove the `Setuptools` integration (`build_sphinx` hook in `setup.py`).
- [#11364](#)¹²³¹: Remove deprecated `sphinx.ext.napoleon.iterators` module.
- [#11365](#)¹²³²: Remove support for the `jsdump` format in `sphinx.search`.
- [#11366](#)¹²³³: Make `locale` a required argument to `sphinx.util.i18n.format_date()`.
- [#11370](#)¹²³⁴: Remove deprecated `sphinx.util.stemmer` module.

¹²²⁴ <https://github.com/sphinx-doc/sphinx/issues/11345>

¹²²⁵ <https://github.com/sphinx-doc/sphinx/issues/11411>

¹²²⁶ <https://docutils.sourceforge.io/RELEASE-NOTES.html#release-0-20-2023-05-04>

¹²²⁷ <https://github.com/sphinx-doc/sphinx/issues/11418>

¹²²⁸ <https://github.com/sphinx-doc/sphinx/issues/11359>

¹²²⁹ <https://github.com/sphinx-doc/sphinx/issues/11360>

¹²³⁰ <https://github.com/sphinx-doc/sphinx/issues/11363>

¹²³¹ <https://github.com/sphinx-doc/sphinx/issues/11364>

¹²³² <https://github.com/sphinx-doc/sphinx/issues/11365>

¹²³³ <https://github.com/sphinx-doc/sphinx/issues/11366>

¹²³⁴ <https://github.com/sphinx-doc/sphinx/issues/11370>

- #11371¹²³⁵: Remove deprecated `sphinx.pycode.ast.parse()` function.
- #11372¹²³⁶: Remove deprecated `sphinx.io.read_doc()` function.
- #11373¹²³⁷: Removed deprecated `sphinx.util.get_matching_files()` function.
- #11378¹²³⁸: Remove deprecated `sphinx.util.docutils.is_html5_writer_available()` function.
- #11379¹²³⁹: Make the `env` argument to `Builder` subclasses required.
- #11380¹²⁴⁰: `autosummary`: Always emit grouped import exceptions.
- #11381¹²⁴¹: Remove deprecated `style` key for HTML templates.
- #11382¹²⁴²: Remove deprecated `sphinx.writers.latex.LaTeXTranslator.docclasses` attribute.
- #11383¹²⁴³: Remove deprecated `sphinx.builders.html.html5_ready` and `sphinx.builders.html.HTMLTranslator` attributes.
- #11385¹²⁴⁴: Remove support for HTML 4 output.

14.22 Release 6.2.1 (released Apr 25, 2023)

Bugs fixed

- #11355¹²⁴⁵: Revert the default type of `nitpick_ignore` and `nitpick_ignore_regex` to `list`.

14.23 Release 6.2.0 (released Apr 23, 2023)

Dipendenze

- Require `Docutils` 0.18.1 or greater.

¹²³⁵ <https://github.com/sphinx-doc/sphinx/issues/11371>

¹²³⁶ <https://github.com/sphinx-doc/sphinx/issues/11372>

¹²³⁷ <https://github.com/sphinx-doc/sphinx/issues/11373>

¹²³⁸ <https://github.com/sphinx-doc/sphinx/issues/11378>

¹²³⁹ <https://github.com/sphinx-doc/sphinx/issues/11379>

¹²⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/11380>

¹²⁴¹ <https://github.com/sphinx-doc/sphinx/issues/11381>

¹²⁴² <https://github.com/sphinx-doc/sphinx/issues/11382>

¹²⁴³ <https://github.com/sphinx-doc/sphinx/issues/11383>

¹²⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/11385>

¹²⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/11355>

Modifiche incompatibili

- LaTeX: removal of some internal TeX `\dimen` registers (not previously publicly documented) as per 5.1.0 code comments in `sphinx.sty`: `\sphinxverbatimsep`, `\sphinxverbatimborder`, `\sphinxshadowsep`, `\sphinxshadowsize`, and `\sphinxshadowrule`. (refs: [#11105](#)¹²⁴⁶)
- Remove `.egg` support from `pycode ModuleAnalyser`; Python eggs are a now-obsolete binary distribution format
- [#11089](#)¹²⁴⁷: Remove deprecated code in `sphinx.builders.linkcheck`. Patch by Daniel Eades
- Remove internal-only `sphinx.locale.setlocale`

Deprecato

- [#11247](#)¹²⁴⁸: Deprecate the legacy `intersphinx_mapping` format
- `sphinx.util.osutil.cd` is deprecated in favour of `contextlib.chdir`.

Funzionalità aggiunte

- [#11277](#)¹²⁴⁹: *autoproperty* allows the return type to be specified as a type comment (e.g., `# type: () -> int`). Patch by Bénédict Tran
- [#10811](#)¹²⁵⁰: Autosummary: extend `__all__` to imported members for template rendering when option `autosummary_ignore_module_all` is set to `False`. Patch by Clement Pinard
- [#11147](#)¹²⁵¹: Add a `content_offset` parameter to `nested_parse_with_titles()`, allowing for correct line numbers during nested parsing. Patch by Jeremy Maitin-Shepard
- Update to Unicode CLDR 42
- Add a `--jobs` synonym for `-j`. Patch by Hugo van Kemenade
- LaTeX: a command `\sphinxbox` for styling text elements with a (possibly rounded) box, optional background color and shadow, has been added. See *Il comando `\sphinxbox`*. (refs: [#11224](#)¹²⁵²)
- LaTeX: add `\sphinxstylenotetitle`, ..., `\sphinxstylewarningtitle`, ..., for an extra layer of mark-up freeing up `\sphinxstrong` for other uses. See *Le Macro*. (refs: [#11267](#)¹²⁵³)

¹²⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/11105>

¹²⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/11089>

¹²⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/11247>

¹²⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/11277>

¹²⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/10811>

¹²⁵¹ <https://github.com/sphinx-doc/sphinx/issues/11147>

¹²⁵² <https://github.com/sphinx-doc/sphinx/issues/11224>

¹²⁵³ <https://github.com/sphinx-doc/sphinx/issues/11267>

- LaTeX: `note`¹²⁵⁴, `hint`¹²⁵⁵, `important`¹²⁵⁶ and `tip`¹²⁵⁷ can now each be styled as the other admonitions, i.e. possibly with a background color, individual border widths and paddings, possibly rounded corners, and optional shadow. See *Chiavi aggiuntive 'sphinxsetup' simili a CSS*. (refs: #11234¹²⁵⁸)
- LaTeX: `admonitions` and `topic`¹²⁵⁹ (and `contents`¹²⁶⁰) directives, and not only `code-block`, support `box-decoration-break=slice`.
- LaTeX: let rounded boxes support up to 4 distinct border-widths (refs: #11243¹²⁶¹)
- LaTeX: new options `noteTextColor`, `noteTeXextras` et al. See *Chiavi aggiuntive 'sphinxsetup' simili a CSS*.
- LaTeX: support elliptical corners in rounded boxes. (refs: #11254¹²⁶²)
- #11150¹²⁶³: Include source location in highlighting warnings, when lexing fails. Patch by Jeremy Maitin-Shepard
- #11281¹²⁶⁴: Support for `imgmath_latex` = 'tectonic' or = 'xelatex'. Patch by Dimitar Dimitrov
- #11109¹²⁶⁵, #9643¹²⁶⁶: Add `python_display_short_literal_types` option for condensed rendering of Literal types.

Bugs fixed

- #11079¹²⁶⁷: LaTeX: figures with align attribute may disappear and strangely impact following lists
- #11093¹²⁶⁸: LaTeX: fix "multiply-defined references" PDF build warnings when one or more reST labels directly precede an `py:module` or `automodule` directive. Patch by B  n  dikt Tran (picnixz)
- #11110¹²⁶⁹: LaTeX: Figures go missing from latex pdf if their files have the same base name and they use a post transform. Patch by aaron-cooper
- LaTeX: fix potential color leak from shadow to border of rounded boxes, if shadow color is set but border color is not

¹²⁵⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#note>

¹²⁵⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#hint>

¹²⁵⁶ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#important>

¹²⁵⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#tip>

¹²⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/11234>

¹²⁵⁹ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

¹²⁶⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

¹²⁶¹ <https://github.com/sphinx-doc/sphinx/issues/11243>

¹²⁶² <https://github.com/sphinx-doc/sphinx/issues/11254>

¹²⁶³ <https://github.com/sphinx-doc/sphinx/issues/11150>

¹²⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/11281>

¹²⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/11109>

¹²⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/9643>

¹²⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/11079>

¹²⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/11093>

¹²⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/11110>

- LaTeX: fix unintended 1pt upwards vertical shift of code blocks frames respective to contents (when using rounded corners)
- #11235¹²⁷⁰: LaTeX: added `\color` in topic (or admonition) contents may cause color leak to the shadow and border at a page break
- #11264¹²⁷¹: LaTeX: missing space before colon after "Voir aussi" for *seealso* directive in French
- #11268¹²⁷²: LaTeX: longtable with left alignment breaks out of current list indentation context in PDF. Thanks to picnixz.
- #11274¹²⁷³: LaTeX: external links are not properly escaped for `\sphinxupquote` compatibility
- #11147¹²⁷⁴: Fix source file/line number info in object description content and in other uses of `nested_parse_with_titles`. Patch by Jeremy Maitin-Shepard.
- #11192¹²⁷⁵: Restore correct parallel search index building. Patch by Jeremy Maitin-Shepard
- Use the new Transifex tx client

Testing

- Fail testing when any Python warnings are emitted
- Migrate remaining `unittest.TestCase` style test functions to `pytest` style
- Remove tests that rely on `setuptools`

14.24 Release 6.1.3 (released Jan 10, 2023)

Bugs fixed

- #11116¹²⁷⁶: Reverted to previous Sphinx 5 node copying method
- #11117¹²⁷⁷: Reverted changes to parallel image processing from Sphinx 6.1.0
- #11119¹²⁷⁸: Suppress `ValueError` in the `linkcheck` builder

¹²⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/11235>

¹²⁷¹ <https://github.com/sphinx-doc/sphinx/issues/11264>

¹²⁷² <https://github.com/sphinx-doc/sphinx/issues/11268>

¹²⁷³ <https://github.com/sphinx-doc/sphinx/issues/11274>

¹²⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/11147>

¹²⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/11192>

¹²⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/11116>

¹²⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/11117>

¹²⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/11119>

14.25 Release 6.1.2 (released Jan 07, 2023)

Bugs fixed

- [#11101](#)¹²⁷⁹: LaTeX: `div.topic_padding` key of `sphinxsetup` documented at 5.1.0 was implemented with name `topic_padding`
- [#11099](#)¹²⁸⁰: LaTeX: `shadowrule` key of `sphinxsetup` causes PDF build to crash since Sphinx 5.1.0
- [#11096](#)¹²⁸¹: LaTeX: `shadowsize` key of `sphinxsetup` causes PDF build to crash since Sphinx 5.1.0
- [#11095](#)¹²⁸²: LaTeX: shadow of `topic`¹²⁸³ and `contents`¹²⁸⁴ boxes not in page margin since Sphinx 5.1.0
- [#11100](#)¹²⁸⁵: Fix copying images when running under parallel mode.

14.26 Release 6.1.1 (released Jan 05, 2023)

Bugs fixed

- [#11091](#)¹²⁸⁶: Fix `util.nodes.apply_source_workaround` for `literal_block` nodes with no source information in the node or the node's parents.

14.27 Release 6.1.0 (released Jan 05, 2023)

Dipendenze

- Adopted the `Ruff`¹²⁸⁷ code linter.

¹²⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/11101>

¹²⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/11099>

¹²⁸¹ <https://github.com/sphinx-doc/sphinx/issues/11096>

¹²⁸² <https://github.com/sphinx-doc/sphinx/issues/11095>

¹²⁸³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

¹²⁸⁴ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#table-of-contents>

¹²⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/11100>

¹²⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/11091>

¹²⁸⁷ <https://github.com/charliermarsh/ruff>

Modifiche incompatibili

- [#10979](#)¹²⁸⁸: gettext: Removed support for pluralisation in `get_translation`. This was unused and complicated other changes to `sphinx.locale`.

Deprecato

- `sphinx.util` functions:
 - Renamed `sphinx.util.typing.stringify()` to `sphinx.util.typing.stringify_annotation()`
 - Moved `sphinx.util.xmlname_checker()` to `sphinx.builders.epub3._XML_NAME_PATTERN`

Moved to `sphinx.util.display`:

- `sphinx.util.status_iterator`
- `sphinx.util.display_chunk`
- `sphinx.util.SkipProgressMessage`
- `sphinx.util.progress_message`

Moved to `sphinx.util.http_date`:

- `sphinx.util.epoch_to_rfc1123`
- `sphinx.util.rfc1123_to_epoch`

Moved to `sphinx.util.exceptions`:

- `sphinx.util.save_traceback`
- `sphinx.util.format_exception_cut_frames`

Funzionalità aggiunte

- Cache doctrees in the build environment during the writing phase.
- Make all writing phase tasks support parallel execution.
- [#11072](#)¹²⁸⁹: Use PEP 604 (X | Y) display conventions for `typing.Optional` and `typing.Optional` types within the Python domain and autodoc.
- [#10700](#)¹²⁹⁰: autodoc: Document `typing.NewType()` types as classes rather than 'data'.
- Cache doctrees between the reading and writing phases.

¹²⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/10979>

¹²⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/11072>

¹²⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/10700>

Bugs fixed

- [#10962](#)¹²⁹¹: HTML: Fix the multi-word key name lookup table.
- Fixed support for Python 3.12 alpha 3 (changes in the `enum` module).
- [#11069](#)¹²⁹²: HTML Theme: Removed outdated "shortcut" link relation keyword.
- [#10952](#)¹²⁹³: Properly terminate parallel processes on programme interruption.
- [#10988](#)¹²⁹⁴: Speed up `TocTree.resolve()` through more efficient copying.
- [#6744](#)¹²⁹⁵: LaTeX: support for `seealso` directive should be via an environment to allow styling.
- [#11074](#)¹²⁹⁶: LaTeX: Can't change `sphinxnote` to use `sphinxheavybox` starting with 5.1.0

14.28 Release 6.0.1 (released Jan 05, 2023)

Dipendenze

- Require Pygments 2.13 or later.

Bugs fixed

- [#10944](#)¹²⁹⁷: `imgmath`: Fix resolving image paths for files in nested folders.

14.29 Release 6.0.0 (released Dec 29, 2022)

Dipendenze

- [#10468](#)¹²⁹⁸: Drop Python 3.6 support
- [#10470](#)¹²⁹⁹: Drop Python 3.7, Docutils 0.14, Docutils 0.15, Docutils 0.16, and Docutils 0.17 support. Patch by Adam Turner

¹²⁹¹ <https://github.com/sphinx-doc/sphinx/issues/10962>

¹²⁹² <https://github.com/sphinx-doc/sphinx/issues/11069>

¹²⁹³ <https://github.com/sphinx-doc/sphinx/issues/10952>

¹²⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/10988>

¹²⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/6744>

¹²⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/11074>

¹²⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/10944>

¹²⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/10468>

¹²⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/10470>

Modifiche incompatibili

- [#7405](#)¹³⁰⁰: Removed the jQuery and underscore.js JavaScript frameworks.

These frameworks are no longer be automatically injected into themes from Sphinx 6.0. If you develop a theme or extension that uses the jQuery, \$, or \$u global objects, you need to update your JavaScript to modern standards, or use the mitigation below.

The first option is to use the [sphinxcontrib.jquery](#)¹³⁰¹ extension, which has been developed by the Sphinx team and contributors. To use this, add `sphinxcontrib.jquery` to the extensions list in `conf.py`, or call `app.setup_extension("sphinxcontrib.jquery")` if you develop a Sphinx theme or extension.

The second option is to manually ensure that the frameworks are present. To re-add jQuery and underscore.js, you will need to copy `jquery.js` and `underscore.js` from the [Sphinx repository](#)¹³⁰² to your static directory, and add the following to your `layout.html`:

```
{%- block scripts %}
    <script src="{{ pathto('_static/jquery.js', resource=True) }}">
    </script>
    <script src="{{ pathto('_static/underscore.js', resource=True) }}">
    </script>
    {{ super() }}
{%- endblock %}
```

Patch by Adam Turner.

- [#10471](#)¹³⁰³, [#10565](#)¹³⁰⁴: Removed deprecated APIs scheduled for removal in Sphinx 6.0. See [API deprecate](#) for details. Patch by Adam Turner.
- [#10901](#)¹³⁰⁵: C Domain: Remove support for parsing pre-v3 style type directives and roles. Also remove associated configuration variables `c_allow_pre_v3` and `c_warn_on_allowed_pre_v3`. Patch by Adam Turner.

Funzionalità aggiunte

- [#10924](#)¹³⁰⁶: LaTeX: adopt better looking defaults for tables and code-blocks. See [latex_table_style](#) and the `pre_border-radius` and `pre_background-TeXcolor` *Chiavi aggiuntive 'sphinxsetup' simili a CSS* for the former defaults and how to re-enact them if desired.

¹³⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/7405>

¹³⁰¹ <https://github.com/sphinx-contrib/jquery/>

¹³⁰² <https://github.com/sphinx-doc/sphinx/tree/v5.3.0/sphinx/themes/basic/static>

¹³⁰³ <https://github.com/sphinx-doc/sphinx/issues/10471>

¹³⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/10565>

¹³⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/10901>

¹³⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/10924>

Bugs fixed

- #10984¹³⁰⁷: LaTeX: Document `latex_additional_files` behavior for files with `.tex` extension.

14.30 Release 5.3.0 (released Oct 16, 2022)

- #10759¹³⁰⁸: LaTeX: add `latex_table_style` and support the 'booktabs', 'borderless', and 'colorrows' styles. (thanks to Stefan Wiehler for initial pull requests #6666¹³⁰⁹, #6671¹³¹⁰)
- #10840¹³¹¹: One can cross-reference including an option value like `:option:--module=foobar`, `:option:--module[=foobar]`, or `:option:--module foobar`. Patch by Martin Liska.
- #10881¹³¹²: autosectionlabel: Record the generated section label to the debug log.
- #10268¹³¹³: Correctly URI-escape image filenames.
- #10887¹³¹⁴: domains: Allow sections in all the content of all object description directives (e.g. `py:function`). Patch by Adam Turner

14.31 Release 5.2.3 (released Sep 30, 2022)

- #10878¹³¹⁵: Fix base64 image embedding in `sphinx.ext.imgmath`
- #10886¹³¹⁶: Add `:nocontentsentry:` flag and global domain table of contents entry control option. Patch by Adam Turner

¹³⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/10984>

¹³⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/10759>

¹³⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/6666>

¹³¹⁰ <https://github.com/sphinx-doc/sphinx/issues/6671>

¹³¹¹ <https://github.com/sphinx-doc/sphinx/issues/10840>

¹³¹² <https://github.com/sphinx-doc/sphinx/issues/10881>

¹³¹³ <https://github.com/sphinx-doc/sphinx/issues/10268>

¹³¹⁴ <https://github.com/sphinx-doc/sphinx/issues/10887>

¹³¹⁵ <https://github.com/sphinx-doc/sphinx/issues/10878>

¹³¹⁶ <https://github.com/sphinx-doc/sphinx/issues/10886>

14.32 Release 5.2.2 (released Sep 27, 2022)

- [#10872](#)¹³¹⁷: Restore link targets for autodoc modules to the top of content. Patch by Dominic Davis-Foster.

14.33 Release 5.2.1 (released Sep 25, 2022)

Bugs fixed

- [#10861](#)¹³¹⁸: Always normalise the pycon3 lexer to pycon.
- Fix using `sphinx.ext.autosummary` with modules containing titles in the module-level docstring.

14.34 Release 5.2.0.post0 (released Sep 24, 2022)

- Recreated source tarballs for Debian maintainers.

14.35 Release 5.2.0 (released Sep 24, 2022)

Dipendenze

- [#10356](#)¹³¹⁹: Sphinx now uses declarative metadata with `pyproject.toml` to create packages, using PyPA's `flit` project as a build backend. Patch by Adam Turner.

Deprecato

- [#10843](#)¹³²⁰: Support for HTML 4 output. Patch by Adam Turner.

¹³¹⁷ <https://github.com/sphinx-doc/sphinx/issues/10872>

¹³¹⁸ <https://github.com/sphinx-doc/sphinx/issues/10861>

¹³¹⁹ <https://github.com/sphinx-doc/sphinx/issues/10356>

¹³²⁰ <https://github.com/sphinx-doc/sphinx/issues/10843>

Funzionalità aggiunte

- [#10738](#)¹³²¹: `napoleon`: Add support for docstring types using 'of', like `type of type`. Example: `tuple of int`.
- [#10286](#)¹³²²: `C++`, support requires clauses not just between the template parameter lists and the declaration.
- [#10755](#)¹³²³: `linkcheck`: Check the source URL of raw directives that use the `url` option.
- [#10781](#)¹³²⁴: Allow `ref` role to be used with definitions and fields.
- [#10717](#)¹³²⁵: `HTML Search`: Increase priority for full title and subtitle matches in search results
- [#10718](#)¹³²⁶: `HTML Search`: Save search result score to the HTML element for debugging
- [#10673](#)¹³²⁷: Make `toctree` accept 'genindex', 'modindex' and 'search' docnames
- [#6316](#)¹³²⁸, [#10804](#)¹³²⁹: Add domain objects to the table of contents. Patch by Adam Turner
- [#6692](#)¹³³⁰: `HTML Search`: Include explicit `index` directive index entries in the search index and search results. Patch by Adam Turner
- [#10816](#)¹³³¹: `imgmath`: Allow embedding images in HTML as base64
- [#10854](#)¹³³²: `HTML Search`: Use browser localStorage for highlight control, stop storing highlight parameters in URL query strings. Patch by Adam Turner.

Bugs fixed

- [#10723](#)¹³³³: `LaTeX`: 5.1.0 has made the 'sphinxsetup' `verbatimwithframe=false` become without effect.
- [#10257](#)¹³³⁴: `C++`, ensure consistent non-specialization template argument representation.
- [#10729](#)¹³³⁵: `C++`, fix parsing of certain non-type template parameter packs.

¹³²¹ <https://github.com/sphinx-doc/sphinx/issues/10738>

¹³²² <https://github.com/sphinx-doc/sphinx/issues/10286>

¹³²³ <https://github.com/sphinx-doc/sphinx/issues/10755>

¹³²⁴ <https://github.com/sphinx-doc/sphinx/issues/10781>

¹³²⁵ <https://github.com/sphinx-doc/sphinx/issues/10717>

¹³²⁶ <https://github.com/sphinx-doc/sphinx/issues/10718>

¹³²⁷ <https://github.com/sphinx-doc/sphinx/issues/10673>

¹³²⁸ <https://github.com/sphinx-doc/sphinx/issues/6316>

¹³²⁹ <https://github.com/sphinx-doc/sphinx/issues/10804>

¹³³⁰ <https://github.com/sphinx-doc/sphinx/issues/6692>

¹³³¹ <https://github.com/sphinx-doc/sphinx/issues/10816>

¹³³² <https://github.com/sphinx-doc/sphinx/issues/10854>

¹³³³ <https://github.com/sphinx-doc/sphinx/issues/10723>

¹³³⁴ <https://github.com/sphinx-doc/sphinx/issues/10257>

¹³³⁵ <https://github.com/sphinx-doc/sphinx/issues/10729>

- [#10715](#)¹³³⁶: Revert [#10520](#)¹³³⁷: "Fix" use of sidebar classes in `agogo.css_t`

14.36 Release 5.1.1 (released Jul 26, 2022)

Bugs fixed

- [#10701](#)¹³³⁸: Fix `ValueError` in the new deque based `sphinx.ext.napoleon` iterator implementation.
- [#10702](#)¹³³⁹: Restore compatability with third-party builders.

14.37 Release 5.1.0 (released Jul 24, 2022)

Dipendenze

- [#10656](#)¹³⁴⁰: Support `Docutils 0.19`¹³⁴¹. Patch by Adam Turner.

Deprecato

- [#10467](#)¹³⁴²: Deprecated `sphinx.util.stemmer` in favour of `snowballstemmer`. Patch by Adam Turner.
- [#9856](#)¹³⁴³: Deprecated `sphinx.ext.napoleon.iterators`.

Funzionalità aggiunte

- [#10444](#)¹³⁴⁴: html theme: Allow specifying multiple CSS files through the `stylesheet` setting in `theme.conf` or by setting `html_style` to an iterable of strings.
- [#10366](#)¹³⁴⁵: std domain: Add support for emphasising placeholders in `option` directives through a new `option_emphasise_placeholders` configuration option.
- [#10439](#)¹³⁴⁶: std domain: Use the repr of some variables when displaying warnings, making whitespace issues easier to identify.

¹³³⁶ <https://github.com/sphinx-doc/sphinx/issues/10715>

¹³³⁷ <https://github.com/sphinx-doc/sphinx/issues/10520>

¹³³⁸ <https://github.com/sphinx-doc/sphinx/issues/10701>

¹³³⁹ <https://github.com/sphinx-doc/sphinx/issues/10702>

¹³⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/10656>

¹³⁴¹ <https://docutils.sourceforge.io/RELEASE-NOTES.html#release-0-19-2022-07-05>

¹³⁴² <https://github.com/sphinx-doc/sphinx/issues/10467>

¹³⁴³ <https://github.com/sphinx-doc/sphinx/issues/9856>

¹³⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/10444>

¹³⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/10366>

¹³⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/10439>

- [#10571](#)¹³⁴⁷: quickstart: Reduce content in the generated `conf.py` file. Patch by Pradyun Gedam.
- [#10648](#)¹³⁴⁸: LaTeX: CSS-named-alike additional *'sphinxsetup'* keys allow to configure four separate border-widths, four paddings, four corner radii, a shadow (possibly inset), colours for border, background, shadow for each of the code-block, topic, attention, caution, danger, error and warning directives.
- [#10655](#)¹³⁴⁹: LaTeX: Explain non-standard encoding in `LatinRules.xdy`
- [#10599](#)¹³⁵⁰: HTML Theme: Wrap consecutive footnotes in an `<aside>` element when using Docutils 0.18 or later, to allow for easier styling. This matches the behaviour introduced in Docutils 0.19. Patch by Adam Turner.
- [#10518](#)¹³⁵¹: config: Add `include_patterns` as the opposite of `exclude_patterns`. Patch by Adam Turner.

Bugs fixed

- [#10594](#)¹³⁵²: HTML Theme: field term colons are doubled if using Docutils 0.18+
- [#10596](#)¹³⁵³: Build failure if Docutils version is 0.18 (not 0.18.1) due to missing `Node.findall()`
- [#10506](#)¹³⁵⁴: LaTeX: build error if highlighting inline code role in figure caption (refs: [#10251](#)¹³⁵⁵)
- [#10634](#)¹³⁵⁶: Make `-P (pdb)` option work better with exceptions triggered from events
- [#10550](#)¹³⁵⁷: py domain: Fix spurious whitespace in unparsing various operators (`+`, `-`, `~`, and `**`). Patch by Adam Turner (refs: [#10551](#)¹³⁵⁸).
- [#10460](#)¹³⁵⁹: logging: Always show node source locations as absolute paths.
- HTML Search: HTML tags are displayed as a part of object name
- HTML Search: search snippets should not be folded
- HTML Search: Minor errors are emitted on fetching search snippets
- HTML Search: The markers for header links are shown in the search result

¹³⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/10571>

¹³⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/10648>

¹³⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/10655>

¹³⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/10599>

¹³⁵¹ <https://github.com/sphinx-doc/sphinx/issues/10518>

¹³⁵² <https://github.com/sphinx-doc/sphinx/issues/10594>

¹³⁵³ <https://github.com/sphinx-doc/sphinx/issues/10596>

¹³⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/10506>

¹³⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/10251>

¹³⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/10634>

¹³⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/10550>

¹³⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/10551>

¹³⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/10460>

- [#10520](#)¹³⁶⁰: HTML Theme: Fix use of sidebar classes in `agogo.css_t`.
- [#6679](#)¹³⁶¹: HTML Theme: Fix inclusion of hidden toctrees in the agogo theme.
- [#10566](#)¹³⁶²: HTML Theme: Fix `enable_search_shortcuts` does not work
- [#8686](#)¹³⁶³: LaTeX: Text can fall out of code-block at end of page and leave artifact on next page
- [#10633](#)¹³⁶⁴: LaTeX: user injected `\color` commands in topic or admonition boxes may cause color leaks in PDF due to upstream [framed.sty](#)¹³⁶⁵ bug
- [#10638](#)¹³⁶⁶: LaTeX: framed coloured boxes in highlighted code (e.g. highlighted diffs using Pygments style 'manni') inherit thickness of code-block frame
- [#10647](#)¹³⁶⁷: LaTeX: Only one `\label` is generated for `desc_signature` node even if it has multiple node IDs
- [#10579](#)¹³⁶⁸: i18n: `UnboundLocalError` is raised on translating raw directive
- [#9577](#)¹³⁶⁹, [#10088](#)¹³⁷⁰: py domain: Fix warning for duplicate Python references when using `:any:` and `autodoc`.
- [#10548](#)¹³⁷¹: HTML Search: fix minor summary issues.

14.38 Release 5.0.2 (released Jun 17, 2022)

Funzionalità aggiunte

- [#10523](#)¹³⁷²: HTML Theme: Expose the Docutils's version info tuple as a template variable, `docutils_version_info`. Patch by Adam Turner.

¹³⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/10520>

¹³⁶¹ <https://github.com/sphinx-doc/sphinx/issues/6679>

¹³⁶² <https://github.com/sphinx-doc/sphinx/issues/10566>

¹³⁶³ <https://github.com/sphinx-doc/sphinx/issues/8686>

¹³⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/10633>

¹³⁶⁵ <https://ctan.org/pkg/framed>

¹³⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/10638>

¹³⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/10647>

¹³⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/10579>

¹³⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/9577>

¹³⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/10088>

¹³⁷¹ <https://github.com/sphinx-doc/sphinx/issues/10548>

¹³⁷² <https://github.com/sphinx-doc/sphinx/issues/10523>

Bugs fixed

- [#10538](#)¹³⁷³: autodoc: Inherited class attribute having docstring is documented even if `autodoc_inherit_docstring` is disabled
- [#10509](#)¹³⁷⁴: autosummary: autosummary fails with a shared library
- [#10497](#)¹³⁷⁵: py domain: Failed to resolve strings in Literal. Patch by Adam Turner.
- [#10523](#)¹³⁷⁶: HTML Theme: Fix double brackets on citation references in Docutils 0.18+. Patch by Adam Turner.
- [#10534](#)¹³⁷⁷: Missing CSS for `nav.contents` in Docutils 0.18+. Patch by Adam Turner.

14.39 Release 5.0.1 (released Jun 03, 2022)

Bugs fixed

- [#10498](#)¹³⁷⁸: gettext: `TypeError` is raised when sorting warning messages if a node has no line number. Patch by Adam Turner.
- [#10493](#)¹³⁷⁹: HTML Theme: `topic`¹³⁸⁰ directive is rendered incorrectly with Docutils 0.18. Patch by Adam Turner.
- [#10495](#)¹³⁸¹: `IndexError` is raised for a `kbd` role having a separator. Patch by Adam Turner.

14.40 Release 5.0.0 (released May 30, 2022)

Dipendenze

5.0.0 b1

- [#10164](#)¹³⁸²: Support Docutils 0.18¹³⁸³. Patch by Adam Turner.

¹³⁷³ <https://github.com/sphinx-doc/sphinx/issues/10538>

¹³⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/10509>

¹³⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/10497>

¹³⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/10523>

¹³⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/10534>

¹³⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/10498>

¹³⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/10493>

¹³⁸⁰ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#topic>

¹³⁸¹ <https://github.com/sphinx-doc/sphinx/issues/10495>

¹³⁸² <https://github.com/sphinx-doc/sphinx/issues/10164>

¹³⁸³ <https://docutils.sourceforge.io/RELEASE-NOTES.html#release-0-18-2021-10-26>

Modifiche incompatibili

5.0.0 b1

- [#10031](#)¹³⁸⁴: `autosummary: sphinx.ext.autosummary.import_by_name()` now raises `ImportExceptionGroup` instead of `ImportError` when it failed to import target object. Please handle the exception if your extension uses the function to import Python object. As a workaround, you can disable the behavior via `grouped_exception=False` keyword argument until v7.0.
- [#9962](#)¹³⁸⁵: `texinfo: Customizing styles of emphasized text via @definfoenclose` command was not supported because the command was deprecated since `texinfo 6.8`
- [#2068](#)¹³⁸⁶: `intersphinx_disabled_reftypes` has changed default value from an empty list to `['std:doc']` as avoid too surprising silent intersphinx resolutions. To migrate: either add an explicit inventory name to the references intersphinx should resolve, or explicitly set the value of this configuration variable to an empty list.
- [#10197](#)¹³⁸⁷: `html theme: Reduce body_min_width` setting in basic theme to 360px
- [#9999](#)¹³⁸⁸: `LaTeX: separate terms from their definitions by a CR` (refs: [#9985](#)¹³⁸⁹)
- [#10062](#)¹³⁹⁰: Change the default language to `'en'` if any language is not set in `conf.py`

5.0.0 final

- [#10474](#)¹³⁹¹: `language` does not accept `None` as it value. The default value of `language` becomes to `'en'` now. Patch by Adam Turner and Takeshi KOMIYA.

Deprecato

5.0.0 b1

- [#10028](#)¹³⁹²: `jQuery` and `underscore.js` will no longer be automatically injected into themes from Sphinx 6.0. If you develop a theme or extension that uses the `jQuery`, `$`, or `$u` global objects, you need to update your JavaScript or use the mitigation below.

To re-add `jQuery` and `underscore.js`, you will need to copy `jquery.js` and `underscore.js` from the [Sphinx repository](#)¹³⁹³ to your `static` directory, and add the following to your `layout.html`:

¹³⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/10031>

¹³⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/9962>

¹³⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/2068>

¹³⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/10197>

¹³⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/9999>

¹³⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/9985>

¹³⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/10062>

¹³⁹¹ <https://github.com/sphinx-doc/sphinx/issues/10474>

¹³⁹² <https://github.com/sphinx-doc/sphinx/issues/10028>

¹³⁹³ <https://github.com/sphinx-doc/sphinx/tree/v5.3.0/sphinx/themes/basic/static>

```
{%- block scripts %}
    <script src="{{ pathto('_static/jquery.js', resource=True) }}">
    </script>
    <script src="{{ pathto('_static/underscore.js', resource=True) }}">
    </script>
    {{ super() }}
{%- endblock %}
```

Patch by Adam Turner.

- `setuptools` integration. The `build_sphinx` sub-command for `setup.py` is marked as deprecated to follow the policy of `setuptools` team.
- The `locale` argument of `sphinx.util.i18n:babel_format_date()` becomes required
- The `language` argument of `sphinx.util.i18n:format_date()` becomes required
- `sphinx.builders.html.html5_ready`
- `sphinx.io.read_doc()`
- `sphinx.util.docutils.__version_info__`
- `sphinx.util.docutils.is_html5_writer_available()`
- `sphinx.writers.latex.LaTeXWriter.docclasses`

Funzionalità aggiunte

5.0.0 b1

- [#9075](https://github.com/sphinx-doc/sphinx/issues/9075)¹³⁹⁴: `autodoc`: The default value of `autodoc_typehints_format` is changed to `'smart'`. It will suppress the leading module names of typehints (ex. `io.StringIO` -> `StringIO`).
- [#8417](https://github.com/sphinx-doc/sphinx/issues/8417)¹³⁹⁵: `autodoc`: `:inherited-members:` option now takes multiple classes. It allows to suppress inherited members of several classes on the module at once by specifying the option to `automodule` directive
- [#9792](https://github.com/sphinx-doc/sphinx/issues/9792)¹³⁹⁶: `autodoc`: Add new option for `autodoc_typehints_description_target` to include undocumented return values but not undocumented parameters.
- [#10285](https://github.com/sphinx-doc/sphinx/issues/10285)¹³⁹⁷: `autodoc`: `singledispatch` functions having typehints are not documented
- `autodoc`: `autodoc_typehints_format` now also applies to attributes, data, properties, and type variable bounds.

¹³⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/9075>

¹³⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/8417>

¹³⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/9792>

¹³⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/10285>

- #10258¹³⁹⁸: autosummary: Recognize a documented attribute of a module as non-imported
- #10028¹³⁹⁹: Removed internal usages of JavaScript frameworks (jQuery and underscore.js) and modernised doctools.js and searchtools.js to ECMAScript 2018. Patch by Adam Turner.
- #10302¹⁴⁰⁰: C++, add support for conditional expressions (?:).
- #5157¹⁴⁰¹, #10251¹⁴⁰²: Inline code is able to be highlighted via `role`¹⁴⁰³ directive
- #10337¹⁴⁰⁴: Make sphinx-build faster by caching Publisher object during build. Patch by Adam Turner.

Bugs fixed

5.0.0 b1

- #10200¹⁴⁰⁵: apidoc: Duplicated submodules are shown for modules having both .pyx and .so files. Patch by Adam Turner and Takeshi KOMIYA.
- #10279¹⁴⁰⁶: autodoc: Default values for keyword only arguments in overloaded functions are rendered as a string literal
- #10280¹⁴⁰⁷: autodoc: `autodoc_docstring_signature` unexpectedly generates return value typehint for constructors if docstring has multiple signatures
- #10266¹⁴⁰⁸: autodoc: `autodoc_preserve_defaults` does not work for mixture of keyword only arguments with/without defaults
- #10310¹⁴⁰⁹: autodoc: class methods are not documented when decorated with mocked function
- #10305¹⁴¹⁰: autodoc: Failed to extract optional forward-ref'ed typehints correctly via `autodoc_type_aliases`
- #10421¹⁴¹¹: autodoc: `autodoc_preserve_defaults` doesn't work on class methods
- #10214¹⁴¹²: html: invalid language tag was generated if `language` contains a country code (ex. zh_CN)

¹³⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/10258>

¹³⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/10028>

¹⁴⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/10302>

¹⁴⁰¹ <https://github.com/sphinx-doc/sphinx/issues/5157>

¹⁴⁰² <https://github.com/sphinx-doc/sphinx/issues/10251>

¹⁴⁰³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#role>

¹⁴⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/10337>

¹⁴⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/10200>

¹⁴⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/10279>

¹⁴⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/10280>

¹⁴⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/10266>

¹⁴⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/10310>

¹⁴¹⁰ <https://github.com/sphinx-doc/sphinx/issues/10305>

¹⁴¹¹ <https://github.com/sphinx-doc/sphinx/issues/10421>

¹⁴¹² <https://github.com/sphinx-doc/sphinx/issues/10214>

- #9974¹⁴¹³: html: Updated jQuery version from 3.5.1 to 3.6.0
- #10236¹⁴¹⁴: html search: objects are duplicated in search result
- #9962¹⁴¹⁵: texinfo: Deprecation message for @definfoenclose command on bulding texinfo document
- #10000¹⁴¹⁶: LaTeX: glossary terms with common definition are rendered with too much vertical whitespace
- #10188¹⁴¹⁷: LaTeX: alternating multiply referred footnotes produce a ? in pdf output
- #10363¹⁴¹⁸: LaTeX: make 'howto' title page rule use \linewidth for compatibility with usage of a twocolumn class option
- #10318¹⁴¹⁹: :prepend: option of *literalinclude* directive does not work with :dedent: option

5.0.0 final

- #9575¹⁴²⁰: autodoc: The annotation of return value should not be shown when autodoc_typehints="description"
- #9648¹⁴²¹: autodoc: *args and **kwargs entries are duplicated when autodoc_typehints="description"
- #8180¹⁴²²: autodoc: Docstring metadata ignored for attributes
- #10443¹⁴²³: epub: EPUB builder can't detect the mimetype of .webp file
- #10104¹⁴²⁴: gettext: Duplicated locations are shown if 3rd party extension does not provide correct information
- #10456¹⁴²⁵: py domain: :meta: fields are displayed if docstring contains two or more meta-field
- #9096¹⁴²⁶: sphinx-build: the value of progress bar for paralle build is wrong
- #10110¹⁴²⁷: sphinx-build: exit code is not changed when error is raised on builder-finished event

¹⁴¹³ <https://github.com/sphinx-doc/sphinx/issues/9974>

¹⁴¹⁴ <https://github.com/sphinx-doc/sphinx/issues/10236>

¹⁴¹⁵ <https://github.com/sphinx-doc/sphinx/issues/9962>

¹⁴¹⁶ <https://github.com/sphinx-doc/sphinx/issues/10000>

¹⁴¹⁷ <https://github.com/sphinx-doc/sphinx/issues/10188>

¹⁴¹⁸ <https://github.com/sphinx-doc/sphinx/issues/10363>

¹⁴¹⁹ <https://github.com/sphinx-doc/sphinx/issues/10318>

¹⁴²⁰ <https://github.com/sphinx-doc/sphinx/issues/9575>

¹⁴²¹ <https://github.com/sphinx-doc/sphinx/issues/9648>

¹⁴²² <https://github.com/sphinx-doc/sphinx/issues/8180>

¹⁴²³ <https://github.com/sphinx-doc/sphinx/issues/10443>

¹⁴²⁴ <https://github.com/sphinx-doc/sphinx/issues/10104>

¹⁴²⁵ <https://github.com/sphinx-doc/sphinx/issues/10456>

¹⁴²⁶ <https://github.com/sphinx-doc/sphinx/issues/9096>

¹⁴²⁷ <https://github.com/sphinx-doc/sphinx/issues/10110>

14.41 Release 4.5.0 (released Mar 28, 2022)

Modifiche incompatibili

- [#10112](#)¹⁴²⁸: extlinks: Disable hardcoded links detector by default
- [#9993](#)¹⁴²⁹, [#10177](#)¹⁴³⁰: std domain: Disallow to refer an inline target via *ref* role

Deprecato

- `sphinx.ext.napoleon.docstring.GoogleDocstring._qualify_name()`

Funzionalità aggiunte

- [#10260](#)¹⁴³¹: Enable `FORCE_COLOR` and `NO_COLOR` for terminal colouring
- [#10234](#)¹⁴³²: autosummary: Add "autosummary" CSS class to summary tables
- [#10125](#)¹⁴³³: extlinks: Improve suggestion message for a reference having title
- [#10112](#)¹⁴³⁴: extlinks: Add `extlinks_detect_hardcoded_links` to enable hardcoded links detector feature
- [#9494](#)¹⁴³⁵, [#9456](#)¹⁴³⁶: html search: Add a config variable `html_show_search_summary` to enable/disable the search summaries
- [#9337](#)¹⁴³⁷: HTML theme, add option `enable_search_shortcuts` that enables `/` as a Quick search shortcut and `Esc` shortcut that removes search highlighting.
- [#10107](#)¹⁴³⁸: i18n: Allow to suppress translation warnings by adding `#noqa` comment to the tail of each translation message
- [#10252](#)¹⁴³⁹: C++, support attributes on classes, unions, and enums.
- [#10253](#)¹⁴⁴⁰: *pep* role now generates URLs based on peps.python.org¹⁴⁴¹

¹⁴²⁸ <https://github.com/sphinx-doc/sphinx/issues/10112>

¹⁴²⁹ <https://github.com/sphinx-doc/sphinx/issues/9993>

¹⁴³⁰ <https://github.com/sphinx-doc/sphinx/issues/10177>

¹⁴³¹ <https://github.com/sphinx-doc/sphinx/issues/10260>

¹⁴³² <https://github.com/sphinx-doc/sphinx/issues/10234>

¹⁴³³ <https://github.com/sphinx-doc/sphinx/issues/10125>

¹⁴³⁴ <https://github.com/sphinx-doc/sphinx/issues/10112>

¹⁴³⁵ <https://github.com/sphinx-doc/sphinx/issues/9494>

¹⁴³⁶ <https://github.com/sphinx-doc/sphinx/issues/9456>

¹⁴³⁷ <https://github.com/sphinx-doc/sphinx/issues/9337>

¹⁴³⁸ <https://github.com/sphinx-doc/sphinx/issues/10107>

¹⁴³⁹ <https://github.com/sphinx-doc/sphinx/issues/10252>

¹⁴⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/10253>

¹⁴⁴¹ <https://peps.python.org>

Bugs fixed

- #9876¹⁴⁴²: autodoc: Failed to document an imported class that is built from native binary module
- #10133¹⁴⁴³: autodoc: Crashed when mocked module is used for type annotation
- #10146¹⁴⁴⁴: autodoc: `autodoc_default_options` does not support no-value option
- #9971¹⁴⁴⁵: autodoc: TypeError is raised when the target object is annotated by unhashable object
- #10205¹⁴⁴⁶: extlinks: Failed to compile regexp on checking hardcoded links
- #10277¹⁴⁴⁷: html search: Could not search short words (ex. "use")
- #9529¹⁴⁴⁸: LaTeX: named auto numbered footnote (ex. [#named]) that is referred multiple times was rendered to a question mark
- #9924¹⁴⁴⁹: LaTeX: multi-line `cpp:function` directive has big vertical spacing in Latexpdf
- #10158¹⁴⁵⁰: LaTeX: excessive whitespace since v4.4.0 for undocumented variables/structure members
- #10175¹⁴⁵¹: LaTeX: named footnote reference is linked to an incorrect footnote if the name is also used in the different document
- #10269¹⁴⁵²: manpage: Failed to resolve the title of `ref` cross references
- #10179¹⁴⁵³: i18n: suppress "rST localization" warning
- #10118¹⁴⁵⁴: imgconverter: Unnecessary availability check is called for remote URIs
- #10181¹⁴⁵⁵: napoleon: attributes are displayed like class attributes for google style docstrings when `napoleon_use_ivar` is enabled
- #10122¹⁴⁵⁶: sphinx-build: make.bat does not check the installation of sphinx-build command before showing help

¹⁴⁴² <https://github.com/sphinx-doc/sphinx/issues/9876>

¹⁴⁴³ <https://github.com/sphinx-doc/sphinx/issues/10133>

¹⁴⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/10146>

¹⁴⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/9971>

¹⁴⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/10205>

¹⁴⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/10277>

¹⁴⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/9529>

¹⁴⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/9924>

¹⁴⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/10158>

¹⁴⁵¹ <https://github.com/sphinx-doc/sphinx/issues/10175>

¹⁴⁵² <https://github.com/sphinx-doc/sphinx/issues/10269>

¹⁴⁵³ <https://github.com/sphinx-doc/sphinx/issues/10179>

¹⁴⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/10118>

¹⁴⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/10181>

¹⁴⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/10122>

14.42 Release 4.4.0 (released Jan 17, 2022)

Dipendenze

- #10007¹⁴⁵⁷: Use `importlib_metadata` for python-3.9 or older
- #10007¹⁴⁵⁸: Drop `setuptools`

Funzionalità aggiunte

- #9075¹⁴⁵⁹: autodoc: Add a config variable `autodoc_typehints_format` to suppress the leading module names of typehints of function signatures (ex. `io.StringIO` -> `StringIO`)
- #9831¹⁴⁶⁰: Autosummary now documents only the members specified in a module's `__all__` attribute if `autosummary_ignore_module_all` is set to `False`. The default behaviour is unchanged. Autogen also now supports this behavior with the `--respect-module-all` switch.
- #9555¹⁴⁶¹: autosummary: Improve error messages on failure to load target object
- #9800¹⁴⁶²: extlinks: Emit warning if a hardcoded link is replaceable by an extlink, suggesting a replacement.
- #9961¹⁴⁶³: html: Support nested `<kbd>` HTML elements in other HTML builders
- #10013¹⁴⁶⁴: html: Allow to change the loading method of JS via `loading_method` parameter for `Sphinx.add_js_file()`
- #9551¹⁴⁶⁵: html search: "Hide Search Matches" link removes "highlight" parameter from URL
- #9815¹⁴⁶⁶: html theme: Wrap sidebar components in div to allow customizing their layout via CSS
- #9827¹⁴⁶⁷: i18n: Sort items in glossary by translated terms
- #9899¹⁴⁶⁸: py domain: Allows to specify cross-reference specifier (`.` and `~`) as `:type:` option

¹⁴⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/10007>

¹⁴⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/10007>

¹⁴⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/9075>

¹⁴⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/9831>

¹⁴⁶¹ <https://github.com/sphinx-doc/sphinx/issues/9555>

¹⁴⁶² <https://github.com/sphinx-doc/sphinx/issues/9800>

¹⁴⁶³ <https://github.com/sphinx-doc/sphinx/issues/9961>

¹⁴⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/10013>

¹⁴⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/9551>

¹⁴⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/9815>

¹⁴⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/9827>

¹⁴⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/9899>

- #9894¹⁴⁶⁹: linkcheck: add option `linkcheck_exclude_documents` to disable link checking in matched documents.
- #9793¹⁴⁷⁰: sphinx-build: Allow to use the parallel build feature in macOS on macOS and Python3.8+
- #10055¹⁴⁷¹: sphinx-build: Create directories when `-w` option given
- #9993¹⁴⁷²: std domain: Allow to refer an inline target (ex. `_`target name``) via `ref` role
- #9981¹⁴⁷³: std domain: Strip value part of the option directive from general index
- #9391¹⁴⁷⁴: texinfo: improve variable in `samp` role
- #9578¹⁴⁷⁵: texinfo: Add `texinfo_cross_references` to disable cross references for readability with standalone readers
- #9822¹⁴⁷⁶ (and #9062¹⁴⁷⁷), add new Intersphinx role `external` for explicit lookup in the external projects, without resolving to the local project.

Bugs fixed

- #9866¹⁴⁷⁸: autodoc: doccomment for the imported class was ignored
- #9883¹⁴⁷⁹: autodoc: doccomment for the alias to mocked object was ignored
- #9908¹⁴⁸⁰: autodoc: debug message is shown on building document using NewTypes with Python 3.10
- #9968¹⁴⁸¹: autodoc: instance variables are not shown if `__init__` method has position-only-arguments
- #9194¹⁴⁸²: autodoc: types under the "typing" module are not hyperlinked
- #10009¹⁴⁸³: autodoc: Crashes if target object raises an error on getting docstring
- #10058¹⁴⁸⁴: autosummary: Imported members are not shown when `autodoc_class_signature = 'separated'`

¹⁴⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/9894>

¹⁴⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/9793>

¹⁴⁷¹ <https://github.com/sphinx-doc/sphinx/issues/10055>

¹⁴⁷² <https://github.com/sphinx-doc/sphinx/issues/9993>

¹⁴⁷³ <https://github.com/sphinx-doc/sphinx/issues/9981>

¹⁴⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/9391>

¹⁴⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/9578>

¹⁴⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/9822>

¹⁴⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/9062>

¹⁴⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/9866>

¹⁴⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/9883>

¹⁴⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/9908>

¹⁴⁸¹ <https://github.com/sphinx-doc/sphinx/issues/9968>

¹⁴⁸² <https://github.com/sphinx-doc/sphinx/issues/9194>

¹⁴⁸³ <https://github.com/sphinx-doc/sphinx/issues/10009>

¹⁴⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/10058>

- #9947¹⁴⁸⁵: i18n: topic directive having a bullet list can't be translatable
- #9878¹⁴⁸⁶: mathjax: MathJax configuration is placed after loading MathJax itself
- #9932¹⁴⁸⁷: napoleon: empty "returns" section is generated even if no description
- #9857¹⁴⁸⁸: Generated RFC links use outdated base url
- #9909¹⁴⁸⁹: HTML, prevent line-wrapping in literal text.
- #10061¹⁴⁹⁰: html theme: Configuration values added by themes are not be able to override from conf.py
- #10073¹⁴⁹¹: imgconverter: Unnecessary availability check is called for "data" URIs
- #9925¹⁴⁹²: LaTeX: prohibit also with 'xelatex' line splitting at dashes of inline and parsed literals
- #9944¹⁴⁹³: LaTeX: extra vertical whitespace for some nested declarations
- #9940¹⁴⁹⁴: LaTeX: Multi-function declaration in Python domain has cramped vertical spacing in latexpdf output
- #10015¹⁴⁹⁵: py domain: types under the "typing" module are not hyperlinked defined at info-field-list
- #9390¹⁴⁹⁶: texinfo: Do not emit labels inside footnotes
- #9413¹⁴⁹⁷: xml: Invalid XML was generated when cross referencing python objects
- #9979¹⁴⁹⁸: Error level messages were displayed as warning messages
- #10057¹⁴⁹⁹: Failed to scan documents if the project is placed onto the root directory
- #9636¹⁵⁰⁰: code-block: :dedent: without argument did strip newlines

¹⁴⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/9947>

¹⁴⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/9878>

¹⁴⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/9932>

¹⁴⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/9857>

¹⁴⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/9909>

¹⁴⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/10061>

¹⁴⁹¹ <https://github.com/sphinx-doc/sphinx/issues/10073>

¹⁴⁹² <https://github.com/sphinx-doc/sphinx/issues/9925>

¹⁴⁹³ <https://github.com/sphinx-doc/sphinx/issues/9944>

¹⁴⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/9940>

¹⁴⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/10015>

¹⁴⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/9390>

¹⁴⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/9413>

¹⁴⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/9979>

¹⁴⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/10057>

¹⁵⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/9636>

14.43 Release 4.3.2 (released Dec 19, 2021)

Bugs fixed

- #9917¹⁵⁰¹: C and C++, parse fundamental types no matter the order of simple type specifiers.

14.44 Release 4.3.1 (released Nov 28, 2021)

Funzionalità aggiunte

- #9864¹⁵⁰²: mathjax: Support changing the loading method of MathJax to "defer" via *mathjax_options*

Bugs fixed

- #9838¹⁵⁰³: autodoc: AttributeError is raised on building document for functions decorated by `functools.lru_cache`
- #9879¹⁵⁰⁴: autodoc: AttributeError is raised on building document for an object having invalid `__doc__` attribute
- #9844¹⁵⁰⁵: autodoc: Failed to process a function wrapped with `functools.partial` if *autodoc_preserve_defaults* enabled
- #9872¹⁵⁰⁶: html: Class namespace collision between autodoc signatures and Docutils 0.17
- #9868¹⁵⁰⁷: imgmath: Crashed if the `dvisvgm` command failed to convert equation
- #9864¹⁵⁰⁸: mathjax: Failed to render equations via MathJax v2. The loading method of MathJax is back to "async" method again

¹⁵⁰¹ <https://github.com/sphinx-doc/sphinx/issues/9917>

¹⁵⁰² <https://github.com/sphinx-doc/sphinx/issues/9864>

¹⁵⁰³ <https://github.com/sphinx-doc/sphinx/issues/9838>

¹⁵⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/9879>

¹⁵⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/9844>

¹⁵⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/9872>

¹⁵⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/9868>

¹⁵⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/9864>

14.45 Release 4.3.0 (released Nov 11, 2021)

Dipendenze

- Support Python 3.10

Modifiche incompatibili

- [#9649](#)¹⁵⁰⁹: `searchindex.js`: the embedded data has changed format to allow objects with the same name in different domains.
- [#9672](#)¹⁵¹⁰: The rendering of Python domain declarations is implemented with more Docutils nodes to allow better CSS styling. It may break existing styling.
- [#9672](#)¹⁵¹¹: the signature of `domains.python.PyObject.get_signature_prefix` has changed to return a list of nodes instead of a plain string.
- [#9695](#)¹⁵¹²: `domains.js.JSObject.display_prefix` has been changed into a method `get_display_prefix` which now returns a list of nodes instead of a plain string.
- [#9695](#)¹⁵¹³: The rendering of Javascript domain declarations is implemented with more Docutils nodes to allow better CSS styling. It may break existing styling.
- [#9450](#)¹⁵¹⁴: `mathjax`: Load MathJax via "defer" strategy

Deprecato

- `sphinx.ext.autodoc.AttributeDocumenter._datadescriptor`
- `sphinx.writers.html.HTMLTranslator._fieldlist_row_index`
- `sphinx.writers.html.HTMLTranslator._table_row_index`
- `sphinx.writers.html5.HTML5Translator._fieldlist_row_index`
- `sphinx.writers.html5.HTML5Translator._table_row_index`

¹⁵⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/9649>

¹⁵¹⁰ <https://github.com/sphinx-doc/sphinx/issues/9672>

¹⁵¹¹ <https://github.com/sphinx-doc/sphinx/issues/9672>

¹⁵¹² <https://github.com/sphinx-doc/sphinx/issues/9695>

¹⁵¹³ <https://github.com/sphinx-doc/sphinx/issues/9695>

¹⁵¹⁴ <https://github.com/sphinx-doc/sphinx/issues/9450>

Funzionalità aggiunte

- [#9639](#)¹⁵¹⁵: autodoc: Support asynchronous generator functions
- [#9664](#)¹⁵¹⁶: autodoc: autodoc-process-bases supports to inject reST snippet as a base class
- [#9691](#)¹⁵¹⁷: C, added new info-field retval for `c:function` and `c:macro`.
- C++, added new info-field retval for `cpp:function`.
- [#9618](#)¹⁵¹⁸: i18n: Add `gettext_allow_fuzzy_translations` to allow "fuzzy" messages for translation
- [#9672](#)¹⁵¹⁹: More CSS classes on Python domain descriptions
- [#9695](#)¹⁵²⁰: More CSS classes on Javascript domain descriptions
- [#9683](#)¹⁵²¹: Revert the removal of `add_stylesheet()` API. It will be kept until the Sphinx 6.0 release
- [#2068](#)¹⁵²², add `intersphinx_disabled_reftypes` for disabling intersphinx resolution of cross-references that do not have an explicit inventory specification. Specific types of cross-references can be disabled, e.g., `std:doc` or all cross-references in a specific domain, e.g., `std:*`.
- [#9623](#)¹⁵²³: Allow to suppress "toctree contains reference to excluded document" warnings using `suppress_warnings`

Bugs fixed

- [#9630](#)¹⁵²⁴: autodoc: Failed to build cross references if `primary_domain` is not 'py'
- [#9644](#)¹⁵²⁵: autodoc: Crashed on getting source info from problematic object
- [#9655](#)¹⁵²⁶: autodoc: mocked object having doc comment is warned unexpectedly
- [#9651](#)¹⁵²⁷: autodoc: return type field is not generated even if `autodoc_typehints_description_target` is set to "documented" when its info-field-list contains `:returns:` field

¹⁵¹⁵ <https://github.com/sphinx-doc/sphinx/issues/9639>

¹⁵¹⁶ <https://github.com/sphinx-doc/sphinx/issues/9664>

¹⁵¹⁷ <https://github.com/sphinx-doc/sphinx/issues/9691>

¹⁵¹⁸ <https://github.com/sphinx-doc/sphinx/issues/9618>

¹⁵¹⁹ <https://github.com/sphinx-doc/sphinx/issues/9672>

¹⁵²⁰ <https://github.com/sphinx-doc/sphinx/issues/9695>

¹⁵²¹ <https://github.com/sphinx-doc/sphinx/issues/9683>

¹⁵²² <https://github.com/sphinx-doc/sphinx/issues/2068>

¹⁵²³ <https://github.com/sphinx-doc/sphinx/issues/9623>

¹⁵²⁴ <https://github.com/sphinx-doc/sphinx/issues/9630>

¹⁵²⁵ <https://github.com/sphinx-doc/sphinx/issues/9644>

¹⁵²⁶ <https://github.com/sphinx-doc/sphinx/issues/9655>

¹⁵²⁷ <https://github.com/sphinx-doc/sphinx/issues/9651>

- #9657¹⁵²⁸: autodoc: The base class for a subclass of mocked object is incorrect
- #9607¹⁵²⁹: autodoc: Incorrect base class detection for the subclasses of the generic class
- #9755¹⁵³⁰: autodoc: memory addresses are shown for aliases
- #9752¹⁵³¹: autodoc: Failed to detect type annotation for slots attribute
- #9756¹⁵³²: autodoc: Crashed if classmethod does not have `__func__` attribute
- #9757¹⁵³³: autodoc: `autodoc_inherit_docstrings` does not effect to overridden classmethods
- #9781¹⁵³⁴: autodoc: `autodoc_preserve_defaults` does not support hexadecimal numeric
- #9630¹⁵³⁵: autosummary: Failed to build summary table if `primary_domain` is not 'py'
- #9670¹⁵³⁶: html: Fix download file with special characters
- #9710¹⁵³⁷: html: Wrong styles for even/odd rows in nested tables
- #9763¹⁵³⁸: html: parameter name and its type annotation are not separated in HTML
- #9649¹⁵³⁹: HTML search: when objects have the same name but in different domains, return all of them as result instead of just one.
- #7634¹⁵⁴⁰: intersphinx: references on the file in sub directory are broken
- #9737¹⁵⁴¹: LaTeX: hlist is rendered as a list containing "aggregated" text
- #9678¹⁵⁴²: linkcheck: file extension was shown twice in warnings
- #9697¹⁵⁴³: py domain: An index entry with parens was registered for `py:method` directive with `:property:` option
- #9775¹⁵⁴⁴: py domain: Literal typehint was converted to a cross reference when `autodoc_typehints='description'`
- #9708¹⁵⁴⁵: needs_extension failed to check double-digit version correctly

¹⁵²⁸ <https://github.com/sphinx-doc/sphinx/issues/9657>

¹⁵²⁹ <https://github.com/sphinx-doc/sphinx/issues/9607>

¹⁵³⁰ <https://github.com/sphinx-doc/sphinx/issues/9755>

¹⁵³¹ <https://github.com/sphinx-doc/sphinx/issues/9752>

¹⁵³² <https://github.com/sphinx-doc/sphinx/issues/9756>

¹⁵³³ <https://github.com/sphinx-doc/sphinx/issues/9757>

¹⁵³⁴ <https://github.com/sphinx-doc/sphinx/issues/9781>

¹⁵³⁵ <https://github.com/sphinx-doc/sphinx/issues/9630>

¹⁵³⁶ <https://github.com/sphinx-doc/sphinx/issues/9670>

¹⁵³⁷ <https://github.com/sphinx-doc/sphinx/issues/9710>

¹⁵³⁸ <https://github.com/sphinx-doc/sphinx/issues/9763>

¹⁵³⁹ <https://github.com/sphinx-doc/sphinx/issues/9649>

¹⁵⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/7634>

¹⁵⁴¹ <https://github.com/sphinx-doc/sphinx/issues/9737>

¹⁵⁴² <https://github.com/sphinx-doc/sphinx/issues/9678>

¹⁵⁴³ <https://github.com/sphinx-doc/sphinx/issues/9697>

¹⁵⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/9775>

¹⁵⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/9708>

- [#9688](#)¹⁵⁴⁶: Fix Sphinx patched `code`¹⁵⁴⁷ does not recognize `:class:` option
- [#9733](#)¹⁵⁴⁸: Fix for logging handler flushing warnings in the middle of the docs build
- [#9656](#)¹⁵⁴⁹: Fix warnings without subtype being incorrectly suppressed
- Intersphinx, for unresolved references with an explicit inventory, e.g., `proj:myFunc`, leave the inventory prefix in the unresolved text.

14.46 Release 4.2.0 (released Sep 12, 2021)

Funzionalità aggiunte

- [#9445](#)¹⁵⁵⁰: autodoc: Support class properties
- [#9479](#)¹⁵⁵¹: autodoc: Emit a warning if target is a mocked object
- [#9560](#)¹⁵⁵²: autodoc: Allow to refer NewType instances with module name in Python 3.10 or above
- [#9447](#)¹⁵⁵³: html theme: Expose the version of Sphinx in the form of tuple as a template variable `sphinx_version_tuple`
- [#9594](#)¹⁵⁵⁴: manpage: Suppress the title of man page if description is empty
- [#9445](#)¹⁵⁵⁵: py domain: `py:property` directive supports `:classmethod:` option to describe the class property
- [#9524](#)¹⁵⁵⁶: test: SphinxTestApp can take `builddir` as an argument
- [#9535](#)¹⁵⁵⁷: C and C++, support more fundamental types, including GNU extensions.

¹⁵⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/9688>

¹⁵⁴⁷ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#code>

¹⁵⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/9733>

¹⁵⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/9656>

¹⁵⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/9445>

¹⁵⁵¹ <https://github.com/sphinx-doc/sphinx/issues/9479>

¹⁵⁵² <https://github.com/sphinx-doc/sphinx/issues/9560>

¹⁵⁵³ <https://github.com/sphinx-doc/sphinx/issues/9447>

¹⁵⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/9594>

¹⁵⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/9445>

¹⁵⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/9524>

¹⁵⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/9535>

Bugs fixed

- [#9608](#)¹⁵⁵⁸: apidoc: apidoc does not generate a module definition for implicit namespace package
- [#9504](#)¹⁵⁵⁹: autodoc: generate incorrect reference to the parent class if the target class inherits the class having `_name` attribute
- [#9537](#)¹⁵⁶⁰, [#9589](#)¹⁵⁶¹: autodoc: Some objects under `typing` module are not displayed well with the HEAD of 3.10
- [#9487](#)¹⁵⁶²: autodoc: typehint for `cached_property` is not shown
- [#9509](#)¹⁵⁶³: autodoc: `AttributeError` is raised on failed resolving typehints
- [#9518](#)¹⁵⁶⁴: autodoc: `autodoc_docstring_signature` does not effect to `__init__()` and `__new__()`
- [#9522](#)¹⁵⁶⁵: autodoc: PEP 585 style typehints having arguments (ex. `list[int]`) are not displayed well
- [#9481](#)¹⁵⁶⁶: autosummary: some warnings contain non-existing filenames
- [#9568](#)¹⁵⁶⁷: autosummary: summarise overlined sectioned headings correctly
- [#9600](#)¹⁵⁶⁸: autosummary: Type annotations which contain commas in autosummary table are not removed completely
- [#9481](#)¹⁵⁶⁹: c domain: some warnings contain non-existing filenames
- [#9481](#)¹⁵⁷⁰: cpp domain: some warnings contain non-existing filenames
- [#9456](#)¹⁵⁷¹: html search: abbreviation marks are inserted to the search result if failed to fetch the content of the page
- [#9617](#)¹⁵⁷²: html search: The JS requirement warning is shown if browser is slow
- [#9267](#)¹⁵⁷³: html theme: CSS and JS files added by theme were loaded twice
- [#9585](#)¹⁵⁷⁴: py domain: `:type:` option for `py:property` directive does not create a hyperlink

¹⁵⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/9608>

¹⁵⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/9504>

¹⁵⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/9537>

¹⁵⁶¹ <https://github.com/sphinx-doc/sphinx/issues/9589>

¹⁵⁶² <https://github.com/sphinx-doc/sphinx/issues/9487>

¹⁵⁶³ <https://github.com/sphinx-doc/sphinx/issues/9509>

¹⁵⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/9518>

¹⁵⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/9522>

¹⁵⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/9481>

¹⁵⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/9568>

¹⁵⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/9600>

¹⁵⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/9481>

¹⁵⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/9481>

¹⁵⁷¹ <https://github.com/sphinx-doc/sphinx/issues/9456>

¹⁵⁷² <https://github.com/sphinx-doc/sphinx/issues/9617>

¹⁵⁷³ <https://github.com/sphinx-doc/sphinx/issues/9267>

¹⁵⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/9585>

- [#9576](#)¹⁵⁷⁵: py domain: Literal typehint was converted to a cross reference
- [#9535](#)¹⁵⁷⁶ comment: C++, fix parsing of defaulted function parameters that are function pointers.
- [#9564](#)¹⁵⁷⁷: smartquotes: don't adjust typography for text with language-highlighted `:code:` role.
- [#9512](#)¹⁵⁷⁸: sphinx-build: crashed with the HEAD of Python 3.10

14.47 Release 4.1.2 (released Jul 27, 2021)

Modifiche incompatibili

- [#9435](#)¹⁵⁷⁹: linkcheck: Disable checking automatically generated anchors on github.com (ex. anchors in reST/Markdown documents)

Bugs fixed

- [#9489](#)¹⁵⁸⁰: autodoc: Custom types using `typing.NewType` are not displayed well with the HEAD of 3.10
- [#9490](#)¹⁵⁸¹: autodoc: Some objects under `typing` module are not displayed well with the HEAD of 3.10
- [#9436](#)¹⁵⁸², [#9471](#)¹⁵⁸³: autodoc: crashed if `autodoc_class_signature = "separated"`
- [#9456](#)¹⁵⁸⁴: html search: `html_copy_source` can't control the search summaries
- [#9500](#)¹⁵⁸⁵: LaTeX: Failed to build Japanese document on Windows
- [#9435](#)¹⁵⁸⁶: linkcheck: Failed to check anchors in github.com

¹⁵⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/9576>

¹⁵⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/9535>

¹⁵⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/9564>

¹⁵⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/9512>

¹⁵⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/9435>

¹⁵⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/9489>

¹⁵⁸¹ <https://github.com/sphinx-doc/sphinx/issues/9490>

¹⁵⁸² <https://github.com/sphinx-doc/sphinx/issues/9436>

¹⁵⁸³ <https://github.com/sphinx-doc/sphinx/issues/9471>

¹⁵⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/9456>

¹⁵⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/9500>

¹⁵⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/9435>

14.48 Release 4.1.1 (released Jul 15, 2021)

Dipendenze

- #9434¹⁵⁸⁷: sphinxcontrib-htmlhelp-2.0.0 or above
- #9434¹⁵⁸⁸: sphinxcontrib-serializinghtml-1.1.5 or above

Bugs fixed

- #9438¹⁵⁸⁹: html: HTML logo or Favicon specified as file not being found on output

14.49 Release 4.1.0 (released Jul 12, 2021)

Dipendenze

- Support jinja2-3.0

Deprecato

- The app argument of `sphinx.environment.BuildEnvironment` becomes required
- `sphinx.application.Sphinx.html_theme`
- `sphinx.ext.autosummary._app`
- `sphinx.util.docstrings.extract_metadata()`

Funzionalità aggiunte

- #8107¹⁵⁹⁰: autodoc: Add `class-doc-from` option to `autoclass` directive to control the content of the specific class like `autoclass_content`
- #8588¹⁵⁹¹: autodoc: `autodoc_type_aliases` now supports dotted name. It allows you to define an alias for a class with module name like `foo.bar.BazClass`
- #9175¹⁵⁹²: autodoc: Special member is not documented in the module
- #9195¹⁵⁹³: autodoc: The arguments of `typing.Literal` are wrongly rendered

¹⁵⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/9434>

¹⁵⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/9434>

¹⁵⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/9438>

¹⁵⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/8107>

¹⁵⁹¹ <https://github.com/sphinx-doc/sphinx/issues/8588>

¹⁵⁹² <https://github.com/sphinx-doc/sphinx/issues/9175>

¹⁵⁹³ <https://github.com/sphinx-doc/sphinx/issues/9195>

- #9185¹⁵⁹⁴: autodoc: *autodoc_typehints* allows 'both' setting to allow typehints to be included both in the signature and description
- #4257¹⁵⁹⁵: autodoc: Add *autodoc_class_signature* to separate the class entry and the definition of `__init__()` method
- #8061¹⁵⁹⁶, #9218¹⁵⁹⁷: autodoc: Support variable comment for alias classes
- #3014¹⁵⁹⁸: autodoc: Add *autodoc-process-bases* to modify the base classes of the class definitions
- #9272¹⁵⁹⁹: autodoc: Render enum values for the default argument value better
- #9384¹⁶⁰⁰: autodoc: `autodoc_typehints='none'` now erases typehints for variables, attributes and properties
- #3257¹⁶⁰¹: autosummary: Support instance attributes for classes
- #9358¹⁶⁰²: html: Add "heading" role to the toctree items
- #9225¹⁶⁰³: html: Add span tag to the return typehint of method/function
- #9129¹⁶⁰⁴: html search: Show search summaries when `html_copy_source = False`
- #9307¹⁶⁰⁵: html search: Prevent corrections and completions in search field
- #9120¹⁶⁰⁶: html theme: Eliminate prompt characters of code-block from copyable text
- #9176¹⁶⁰⁷: i18n: Emit a debug message if message catalog file not found under *locale_dirs*
- #9414¹⁶⁰⁸: LaTeX: Add *xeCJKVerbAddon* to default *fvset* config for Chinese documents
- #9016¹⁶⁰⁹: linkcheck: Support checking anchors on github.com
- #9016¹⁶¹⁰: linkcheck: Add a new event *linkcheck-process-uri* to modify URIs before checking hyperlinks
- #6525¹⁶¹¹: linkcheck: Add *linkcheck_allowed_redirects* to mark hyperlinks that are redirected to expected URLs as "working"

¹⁵⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/9185>

¹⁵⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/4257>

¹⁵⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/8061>

¹⁵⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/9218>

¹⁵⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/3014>

¹⁵⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/9272>

¹⁶⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/9384>

¹⁶⁰¹ <https://github.com/sphinx-doc/sphinx/issues/3257>

¹⁶⁰² <https://github.com/sphinx-doc/sphinx/issues/9358>

¹⁶⁰³ <https://github.com/sphinx-doc/sphinx/issues/9225>

¹⁶⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/9129>

¹⁶⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/9307>

¹⁶⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/9120>

¹⁶⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/9176>

¹⁶⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/9414>

¹⁶⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/9016>

¹⁶¹⁰ <https://github.com/sphinx-doc/sphinx/issues/9016>

¹⁶¹¹ <https://github.com/sphinx-doc/sphinx/issues/6525>

- #1874¹⁶¹²: py domain: Support union types using | in info-field-list
- #9268¹⁶¹³: py domain: `python_use_unqualified_type_names` supports type field in info-field-list
- #9097¹⁶¹⁴: Optimize the parallel build
- #9131¹⁶¹⁵: Add `nitpick_ignore_regex` to ignore nitpicky warnings using regular expressions
- #9174¹⁶¹⁶: Add `Sphinx.set_html_assets_policy` to tell extensions to include HTML assets in all the pages. Extensions can check this via `Sphinx.registry.html_assets_policy`
- C++, add support for
 - inline variables,
 - consteval functions,
 - constexpr variables,
 - char8_t,
 - explicit(<constant expression>) specifier,
 - digit separators in literals, and
 - constraints in placeholder type specifiers, aka. adjective syntax (e.g., Sortable auto &v).
- C, add support for digit separators in literals.
- #9166¹⁶¹⁷: LaTeX: support containers in LaTeX output

Bugs fixed

- #8872¹⁶¹⁸: autodoc: stacked singledispatches are wrongly rendered
- #8597¹⁶¹⁹: autodoc: a docstring having metadata only should be treated as undocumented
- #9185¹⁶²⁰: autodoc: typehints for overloaded functions and methods are inaccurate
- #9250¹⁶²¹: autodoc: The inherited method not having docstring is wrongly parsed

¹⁶¹² <https://github.com/sphinx-doc/sphinx/issues/1874>

¹⁶¹³ <https://github.com/sphinx-doc/sphinx/issues/9268>

¹⁶¹⁴ <https://github.com/sphinx-doc/sphinx/issues/9097>

¹⁶¹⁵ <https://github.com/sphinx-doc/sphinx/issues/9131>

¹⁶¹⁶ <https://github.com/sphinx-doc/sphinx/issues/9174>

¹⁶¹⁷ <https://github.com/sphinx-doc/sphinx/issues/9166>

¹⁶¹⁸ <https://github.com/sphinx-doc/sphinx/issues/8872>

¹⁶¹⁹ <https://github.com/sphinx-doc/sphinx/issues/8597>

¹⁶²⁰ <https://github.com/sphinx-doc/sphinx/issues/9185>

¹⁶²¹ <https://github.com/sphinx-doc/sphinx/issues/9250>

- [#9283](#)¹⁶²²: autodoc: autoattribute directive failed to generate document for an attribute not having any comment
- [#9364](#)¹⁶²³: autodoc: single element tuple on the default argument value is wrongly rendered
- [#9362](#)¹⁶²⁴: autodoc: AttributeError is raised on processing a subclass of Tuple[()]
- [#9404](#)¹⁶²⁵: autodoc: TypeError is raised on processing dict-like object (not a class) via autoclass directive
- [#9317](#)¹⁶²⁶: html: Pushing left key causes visiting the next page at the first page
- [#9381](#)¹⁶²⁷: html: URL for html_favicon and html_log does not work
- [#9270](#)¹⁶²⁸: html theme : pyramid theme generates incorrect logo links
- [#9217](#)¹⁶²⁹: manpage: The name of manpage directory that is generated by *man_make_section_directory* is not correct
- [#9350](#)¹⁶³⁰: manpage: Fix font isn't reset after keyword at the top of samp role
- [#9306](#)¹⁶³¹: Linkcheck reports broken link when remote server closes the connection on HEAD request
- [#9280](#)¹⁶³²: py domain: "exceptions" module is not displayed
- [#9418](#)¹⁶³³: py domain: a Callable annotation with no parameters (e.g. Callable[[], None]) will be rendered with a bracket missing (Callable[], None])
- [#9319](#)¹⁶³⁴: quickstart: Make sphinx-quickstart exit when conf.py already exists
- [#9387](#)¹⁶³⁵: xml: XML Builder ignores custom visitors
- [#9224](#)¹⁶³⁶: :param: and :type: fields does not support a type containing whitespace (ex. Dict[str, str])
- [#8945](#)¹⁶³⁷: when transforming typed fields, call the specified role instead of making an single xref. For C and C++, use the expr role for typed fields.

¹⁶²² <https://github.com/sphinx-doc/sphinx/issues/9283>

¹⁶²³ <https://github.com/sphinx-doc/sphinx/issues/9364>

¹⁶²⁴ <https://github.com/sphinx-doc/sphinx/issues/9362>

¹⁶²⁵ <https://github.com/sphinx-doc/sphinx/issues/9404>

¹⁶²⁶ <https://github.com/sphinx-doc/sphinx/issues/9317>

¹⁶²⁷ <https://github.com/sphinx-doc/sphinx/issues/9381>

¹⁶²⁸ <https://github.com/sphinx-doc/sphinx/issues/9270>

¹⁶²⁹ <https://github.com/sphinx-doc/sphinx/issues/9217>

¹⁶³⁰ <https://github.com/sphinx-doc/sphinx/issues/9350>

¹⁶³¹ <https://github.com/sphinx-doc/sphinx/issues/9306>

¹⁶³² <https://github.com/sphinx-doc/sphinx/issues/9280>

¹⁶³³ <https://github.com/sphinx-doc/sphinx/issues/9418>

¹⁶³⁴ <https://github.com/sphinx-doc/sphinx/issues/9319>

¹⁶³⁵ <https://github.com/sphinx-doc/sphinx/issues/9387>

¹⁶³⁶ <https://github.com/sphinx-doc/sphinx/issues/9224>

¹⁶³⁷ <https://github.com/sphinx-doc/sphinx/issues/8945>

14.50 Release 4.0.3 (released Jul 05, 2021)

Funzionalità aggiunte

- C, add C23 keywords `_Decimal32`, `_Decimal64`, and `_Decimal128`.
- #9354¹⁶³⁸: C, add `c_extra_keywords` to allow user-defined keywords during parsing.
- Revert the removal of `sphinx.util:force_decode()` to become some 3rd party extensions available again during 5.0

Bugs fixed

- #9330¹⁶³⁹: changeset domain: `versionchanged` with contents being a list will cause error during pdf build
- #9313¹⁶⁴⁰: LaTeX: complex table with merged cells broken since 4.0
- #9305¹⁶⁴¹: LaTeX: backslash may cause Improper discretionary list pdf build error with Japanese engines
- #9354¹⁶⁴²: C, remove special macro names from the keyword list. See also `c_extra_keywords`.
- #9322¹⁶⁴³: KeyError is raised on PropagateDescDomain transform

14.51 Release 4.0.2 (released May 20, 2021)

Dipendenze

- #9216¹⁶⁴⁴: Support jinja2-3.0

¹⁶³⁸ <https://github.com/sphinx-doc/sphinx/issues/9354>

¹⁶³⁹ <https://github.com/sphinx-doc/sphinx/issues/9330>

¹⁶⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/9313>

¹⁶⁴¹ <https://github.com/sphinx-doc/sphinx/issues/9305>

¹⁶⁴² <https://github.com/sphinx-doc/sphinx/issues/9354>

¹⁶⁴³ <https://github.com/sphinx-doc/sphinx/issues/9322>

¹⁶⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/9216>

Modifiche incompatibili

- #9222¹⁶⁴⁵: Update Underscore.js to 1.13.1
- #9217¹⁶⁴⁶: manpage: Stop creating a section directory on build manpage by default (see *man_make_section_directory*)

Bugs fixed

- #9210¹⁶⁴⁷: viewcode: crashed if non importable modules found on parallel build
- #9240¹⁶⁴⁸: Unknown node error for pending_xref_condition is raised if an extension that does not support the node installs a missing-reference handler

14.52 Release 4.0.1 (released May 11, 2021)

Bugs fixed

- #9189¹⁶⁴⁹: autodoc: crashed when ValueError is raised on generating signature from a property of the class
- #9188¹⁶⁵⁰: autosummary: warning is emitted if list value is set to autosummary_generate
- #8380¹⁶⁵¹: html search: tags for search result are broken
- #9198¹⁶⁵²: i18n: Babel emits errors when running compile_catalog
- #9205¹⁶⁵³: py domain: The :canonical: option causes "more than one target for cross-reference" warning
- #9201¹⁶⁵⁴: websupport: UndefinedError is raised: 'css_tag' is undefined

¹⁶⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/9222>

¹⁶⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/9217>

¹⁶⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/9210>

¹⁶⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/9240>

¹⁶⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/9189>

¹⁶⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/9188>

¹⁶⁵¹ <https://github.com/sphinx-doc/sphinx/issues/8380>

¹⁶⁵² <https://github.com/sphinx-doc/sphinx/issues/9198>

¹⁶⁵³ <https://github.com/sphinx-doc/sphinx/issues/9205>

¹⁶⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/9201>

14.53 Release 4.0.0 (released May 09, 2021)

Dipendenze

4.0.0b1

- Drop python 3.5 support
- Drop Docutils 0.12 and 0.13 support
- LaTeX: add `tex-gyre` font dependency

4.0.0b2

- Support Docutils 0.17. Please notice it changes the output of HTML builder. Some themes do not support it, and you need to update your custom CSS to upgrade it.

Modifiche incompatibili

4.0.0b1

- [#8539](#)¹⁶⁵⁵: `autodoc`: `info-field-list` is generated into the class description when `autodoc_typehints='description'` and `autoclass_content='class'` set
- [#8898](#)¹⁶⁵⁶: `extlinks`: `"%s"` becomes required keyword in the link caption string
- `domain`: The `Index` class becomes subclasses of `abc.ABC` to indicate methods that must be overridden in the concrete classes
- [#4826](#)¹⁶⁵⁷: `py domain`: The structure of python objects is changed. A boolean value is added to indicate that the python object is canonical one
- [#7425](#)¹⁶⁵⁸: `MathJax`: The MathJax was changed from 2 to 3. Users using a custom MathJax configuration may have to set the old MathJax path or update their configuration for version 3. See [sphinx.ext.mathjax](#).
- [#7784](#)¹⁶⁵⁹: `i18n`: The msgid for alt text of image is changed
- [#5560](#)¹⁶⁶⁰: `napoleon`: `napoleon_use_param` also affect "other parameters" section
- [#7996](#)¹⁶⁶¹: `manpage`: Make a section directory on build manpage by default (see [man_make_section_directory](#))
- [#7849](#)¹⁶⁶²: `html`: Change the default setting of `html_codeblock_linenos_style` to `'inline'`

¹⁶⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/8539>

¹⁶⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/8898>

¹⁶⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/4826>

¹⁶⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/7425>

¹⁶⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/7784>

¹⁶⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/5560>

¹⁶⁶¹ <https://github.com/sphinx-doc/sphinx/issues/7996>

¹⁶⁶² <https://github.com/sphinx-doc/sphinx/issues/7849>

- [#8380¹⁶⁶³](#): html search: search results are wrapped with `<p>` instead of `<div>`
- html theme: Move a script tag for `documentation_options.js` in `basic/layout.html` to `script_files` variable
- html theme: Move CSS tags in `basic/layout.html` to `css_files` variable
- [#8915¹⁶⁶⁴](#): html theme: Emit a warning for `sphinx_rtd_theme` 0.2.4 or older
- [#8508¹⁶⁶⁵](#): LaTeX: `uplatex` becomes a default setting of `latex_engine` for Japanese documents
- [#5977¹⁶⁶⁶](#): py domain: `:var:`, `:cvar:` and `:ivar:` fields do not create cross-references
- [#4550¹⁶⁶⁷](#): The `align` attribute of `figure` and `table` nodes becomes `None` by default instead of `'default'`
- [#8769¹⁶⁶⁸](#): LaTeX refactoring: split `sphinx.sty` into multiple files and rename some auxiliary files created in `latex` build output repertory
- [#8937¹⁶⁶⁹](#): Use explicit title instead of `<no title>`
- [#8487¹⁶⁷⁰](#): The `:file:` option for `csv-table` directive now recognizes an absolute path as a relative path from source directory

4.0.0b2

- [#9023¹⁶⁷¹](#): Change the CSS classes on `cpp:expr` and `cpp:texpr`.

Deprecato

- `html_codeblock_linenos_style`
- `favicon` and `logo` variable in HTML templates
- `sphinx.directives.patches.CSVTable`
- `sphinx.directives.patches.ListTable`
- `sphinx.directives.patches.RSTTable`
- `sphinx.ext.autodoc.directive.DocumenterBridge.filename_set`
- `sphinx.ext.autodoc.directive.DocumenterBridge.warn()`
- `sphinx.registry.SphinxComponentRegistry.get_source_input()`
- `sphinx.registry.SphinxComponentRegistry.source_inputs`

¹⁶⁶³ <https://github.com/sphinx-doc/sphinx/issues/8380>

¹⁶⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/8915>

¹⁶⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/8508>

¹⁶⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/5977>

¹⁶⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/4550>

¹⁶⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/8769>

¹⁶⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/8937>

¹⁶⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/8487>

¹⁶⁷¹ <https://github.com/sphinx-doc/sphinx/issues/9023>

- `sphinx.transforms.FigureAligner`
- `sphinx.util.pycompat.convert_with_2to3()`
- `sphinx.util.pycompat.execfile_()`
- `sphinx.util.smartypants`
- `sphinx.util.typing.DirectiveOption`

Funzionalità aggiunte

4.0.0b1

- [#8924](#)¹⁶⁷²: autodoc: Support bound argument for `TypeVar`
- [#7383](#)¹⁶⁷³: autodoc: Support typehints for properties
- [#5603](#)¹⁶⁷⁴: autodoc: Allow to refer to a python class using its canonical name when the class has two different names; a canonical name and an alias name
- [#8539](#)¹⁶⁷⁵: autodoc: Add `autodoc_typehints_description_target` to control the behavior of `autodoc_typehints=description`
- [#8841](#)¹⁶⁷⁶: autodoc: `autodoc_docstring_signature` will continue to look for multiple signature lines without backslash character
- [#7549](#)¹⁶⁷⁷: autosummary: Enable `autosummary_generate` by default
- [#8898](#)¹⁶⁷⁸: extlinks: Allow `%s` in link caption string
- [#4826](#)¹⁶⁷⁹: py domain: Add `:canonical:` option to python directives to describe the location where the object is defined
- [#7199](#)¹⁶⁸⁰: py domain: Add `python_use_unqualified_type_names` to suppress the module name of the python reference if it can be resolved (experimental)
- [#7068](#)¹⁶⁸¹: py domain: Add `py:property` directive to describe a property
- [#7784](#)¹⁶⁸²: i18n: The alt text for image is translated by default (without `gettext_additional_targets` setting)
- [#2018](#)¹⁶⁸³: html: `html_favicon` and `html_logo` now accept URL for the image

¹⁶⁷² <https://github.com/sphinx-doc/sphinx/issues/8924>

¹⁶⁷³ <https://github.com/sphinx-doc/sphinx/issues/7383>

¹⁶⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/5603>

¹⁶⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/8539>

¹⁶⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/8841>

¹⁶⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/7549>

¹⁶⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/8898>

¹⁶⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/4826>

¹⁶⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/7199>

¹⁶⁸¹ <https://github.com/sphinx-doc/sphinx/issues/7068>

¹⁶⁸² <https://github.com/sphinx-doc/sphinx/issues/7784>

¹⁶⁸³ <https://github.com/sphinx-doc/sphinx/issues/2018>

- [#8070¹⁶⁸⁴](#): html search: Support searching for 2characters word
- [#9036¹⁶⁸⁵](#): html theme: Allow to inherit the search page
- [#8938¹⁶⁸⁶](#): imgconverter: Show the error of the command availability check
- [#7830¹⁶⁸⁷](#): Add debug logs for change detection of sources and templates
- [#8201¹⁶⁸⁸](#): Emit a warning if toctree contains duplicated entries
- [#8326¹⁶⁸⁹](#): master_doc is now renamed to *root_doc*
- [#8942¹⁶⁹⁰](#): C++, add support for the C++20 spaceship operator, `<=>`.
- [#7199¹⁶⁹¹](#): A new node, `sphinx.addnodes.pending_xref_condition` has been added. It can be used to choose appropriate content of the reference by conditions.

4.0.0b2

- [#8818¹⁶⁹²](#): autodoc: Super class having Any arguments causes nit-picky warning
- [#9095¹⁶⁹³](#): autodoc: TypeError is raised on processing broken metaclass
- [#9110¹⁶⁹⁴](#): autodoc: metadata of GenericAlias is not rendered as a reference in py37+
- [#9098¹⁶⁹⁵](#): html: copy-range protection for doctests doesn't work in Safari
- [#9103¹⁶⁹⁶](#): LaTeX: imgconverter: conversion runs even if not needed
- [#8127¹⁶⁹⁷](#): py domain: Ellipsis in info-field-list causes nit-picky warning
- [#9121¹⁶⁹⁸](#): py domain: duplicated warning is emitted when both canonical and its alias objects are defined on the document
- [#9023¹⁶⁹⁹](#): More CSS classes on domain descriptions, see *Classi dei nodi Doctree aggiunte da Sphinx* for details.
- [#8195¹⁷⁰⁰](#): mathjax: Rename *mathjax_config* to *mathjax2_config* and add *mathjax3_config*

¹⁶⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/8070>

¹⁶⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/9036>

¹⁶⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/8938>

¹⁶⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/7830>

¹⁶⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/8201>

¹⁶⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/8326>

¹⁶⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/8942>

¹⁶⁹¹ <https://github.com/sphinx-doc/sphinx/issues/7199>

¹⁶⁹² <https://github.com/sphinx-doc/sphinx/issues/8818>

¹⁶⁹³ <https://github.com/sphinx-doc/sphinx/issues/9095>

¹⁶⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/9110>

¹⁶⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/9098>

¹⁶⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/9103>

¹⁶⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/8127>

¹⁶⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/9121>

¹⁶⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/9023>

¹⁷⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/8195>

Bugs fixed

4.0.0b1

- [#8917¹⁷⁰¹](#): autodoc: Raises a warning if function has wrong `__globals__` value
- [#8415¹⁷⁰²](#): autodoc: a `TypeVar` imported from other module is not resolved (in Python 3.7 or above)
- [#8992¹⁷⁰³](#): autodoc: Failed to resolve types.`TracebackType` type annotation
- [#8905¹⁷⁰⁴](#): html: `html_add_permaLinks=None` and `html_add_permaLinks=""` are ignored
- [#8380¹⁷⁰⁵](#): html search: Paragraphs in search results are not identified as `<p>`
- [#8915¹⁷⁰⁶](#): html theme: The translation of `sphinx_rtd_theme` does not work
- [#8342¹⁷⁰⁷](#): Emit a warning if a unknown domain is given for directive or role (ex. `:unknown:doc:`)
- [#7241¹⁷⁰⁸](#): LaTeX: No wrapping for `cpp:enumerator`
- [#8711¹⁷⁰⁹](#): LaTeX: backticks in code-blocks trigger latexpdf build warning (and font change) with late TeXLive 2019
- [#8253¹⁷¹⁰](#): LaTeX: Figures with no size defined get overscaled (compared to images with size explicitly set in pixels) (fixed for `'pdflatex'/'lualatex'` only)
- [#8881¹⁷¹¹](#): LaTeX: The depth of bookmarks panel in PDF is not enough for navigation
- [#8874¹⁷¹²](#): LaTeX: the fix to two minor Pygments LaTeXFormatter output issues ignore Pygments style
- [#8925¹⁷¹³](#): LaTeX: 3.5.0 `verbatimmaxunderfull` setting does not work as expected
- [#8980¹⁷¹⁴](#): LaTeX: missing line break in `\pysigline`
- [#8995¹⁷¹⁵](#): LaTeX: legacy `\pysiglinewithargsret` does not compute correctly available horizontal space and should use a ragged right style
- [#9009¹⁷¹⁶](#): LaTeX: "release" value with underscore leads to invalid LaTeX

¹⁷⁰¹ <https://github.com/sphinx-doc/sphinx/issues/8917>

¹⁷⁰² <https://github.com/sphinx-doc/sphinx/issues/8415>

¹⁷⁰³ <https://github.com/sphinx-doc/sphinx/issues/8992>

¹⁷⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/8905>

¹⁷⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/8380>

¹⁷⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/8915>

¹⁷⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/8342>

¹⁷⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/7241>

¹⁷⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/8711>

¹⁷¹⁰ <https://github.com/sphinx-doc/sphinx/issues/8253>

¹⁷¹¹ <https://github.com/sphinx-doc/sphinx/issues/8881>

¹⁷¹² <https://github.com/sphinx-doc/sphinx/issues/8874>

¹⁷¹³ <https://github.com/sphinx-doc/sphinx/issues/8925>

¹⁷¹⁴ <https://github.com/sphinx-doc/sphinx/issues/8980>

¹⁷¹⁵ <https://github.com/sphinx-doc/sphinx/issues/8995>

¹⁷¹⁶ <https://github.com/sphinx-doc/sphinx/issues/9009>

- #8911¹⁷¹⁷: C++: remove the longest matching prefix in `cpp_index_common_prefix` instead of the first that matches.
- C, properly reject function declarations when a keyword is used as parameter name.
- #8933¹⁷¹⁸: viewcode: Failed to create back-links on parallel build
- #8960¹⁷¹⁹: C and C++, fix rendering of (member) function pointer types in function parameter lists.
- C++, fix linking of names in array declarators, pointer to member (function) declarators, and in the argument to `sizeof`...
- C, fix linking of names in array declarators.

4.0.0b2

- C, C++, fix `KeyError` when an `alias` directive is the first C/C++ directive in a file with another C/C++ directive later.

4.0.0b3

- #9167¹⁷²⁰: html: Failed to add CSS files to the specific page

14.54 Release 3.5.5 (in development)

14.55 Release 3.5.4 (released Apr 11, 2021)

Dipendenze

- #9071¹⁷²¹: Restrict Docutils to 0.16

Bugs fixed

- #9078¹⁷²²: autodoc: Async staticmethods and classmethods are considered as non async coroutine-functions with Python3.10
- #8870¹⁷²³, #9001¹⁷²⁴, #9051¹⁷²⁵: html theme: The style are not applied with Docutils 0.17
 - toctree captions
 - The content of `sidebar` directive

¹⁷¹⁷ <https://github.com/sphinx-doc/sphinx/issues/8911>

¹⁷¹⁸ <https://github.com/sphinx-doc/sphinx/issues/8933>

¹⁷¹⁹ <https://github.com/sphinx-doc/sphinx/issues/8960>

¹⁷²⁰ <https://github.com/sphinx-doc/sphinx/issues/9167>

¹⁷²¹ <https://github.com/sphinx-doc/sphinx/issues/9071>

¹⁷²² <https://github.com/sphinx-doc/sphinx/issues/9078>

¹⁷²³ <https://github.com/sphinx-doc/sphinx/issues/8870>

¹⁷²⁴ <https://github.com/sphinx-doc/sphinx/issues/9001>

¹⁷²⁵ <https://github.com/sphinx-doc/sphinx/issues/9051>

– figures

14.56 Release 3.5.3 (released Mar 20, 2021)

Funzionalità aggiunte

- #8959¹⁷²⁶: using UNIX path separator in image directive confuses Sphinx on Windows

14.57 Release 3.5.2 (released Mar 06, 2021)

Bugs fixed

- #8943¹⁷²⁷: i18n: Crashed by broken translation messages in ES, EL and HR
- #8936¹⁷²⁸: LaTeX: A custom LaTeX builder fails with unknown node error
- #8952¹⁷²⁹: Exceptions raised in a Directive cause parallel builds to hang

14.58 Release 3.5.1 (released Feb 16, 2021)

Bugs fixed

- #8883¹⁷³⁰: autodoc: AttributeError is raised on assigning `__annotations__` on read-only class
- #8884¹⁷³¹: html: minified js stemmers not included in the distributed package
- #8885¹⁷³²: html: AttributeError is raised if CSS/JS files are installed via `html_context`
- #8880¹⁷³³: viewcode: ExtensionError is raised on incremental build after unparsable python module found

¹⁷²⁶ <https://github.com/sphinx-doc/sphinx/issues/8959>

¹⁷²⁷ <https://github.com/sphinx-doc/sphinx/issues/8943>

¹⁷²⁸ <https://github.com/sphinx-doc/sphinx/issues/8936>

¹⁷²⁹ <https://github.com/sphinx-doc/sphinx/issues/8952>

¹⁷³⁰ <https://github.com/sphinx-doc/sphinx/issues/8883>

¹⁷³¹ <https://github.com/sphinx-doc/sphinx/issues/8884>

¹⁷³² <https://github.com/sphinx-doc/sphinx/issues/8885>

¹⁷³³ <https://github.com/sphinx-doc/sphinx/issues/8880>

14.59 Release 3.5.0 (released Feb 14, 2021)

Dipendenze

- LaTeX: `multicol` (it is anyhow a required part of the official latex2e base distribution)

Modifiche incompatibili

- Update Underscore.js to 1.12.0
- #6550¹⁷³⁴: `html`: The config variable `html_add_permalink` is replaced by `html_permalink` and `html_permalink_icon`

Deprecato

- `nodo pending_xref` per l'estensione `viewcode`
- `sphinx.builders.linkcheck.CheckExternalLinksBuilder.anchors_ignore`
- `sphinx.builders.linkcheck.CheckExternalLinksBuilder.auth`
- `sphinx.builders.linkcheck.CheckExternalLinksBuilder.broken`
- `sphinx.builders.linkcheck.CheckExternalLinksBuilder.good`
- `sphinx.builders.linkcheck.CheckExternalLinksBuilder.redirected`
- `sphinx.builders.linkcheck.CheckExternalLinksBuilder.rqueue`
- `sphinx.builders.linkcheck.CheckExternalLinksBuilder.to_ignore`
- `sphinx.builders.linkcheck.CheckExternalLinksBuilder.workers`
- `sphinx.builders.linkcheck.CheckExternalLinksBuilder.wqueue`
- `sphinx.builders.linkcheck.node_line_or_0()`
- `sphinx.ext.autodoc.AttributeDocumenter.isinstanceattribute()`
- `sphinx.ext.autodoc.directive.DocumenterBridge.reporter`
- `sphinx.ext.autodoc.importer.get_module_members()`
- `sphinx.ext.autosummary.generate._simple_info()`
- `sphinx.ext.autosummary.generate._simple_warn()`
- `sphinx.writers.html.HTMLTranslator.permalink_text`
- `sphinx.writers.html5.HTML5Translator.permalink_text`

¹⁷³⁴ <https://github.com/sphinx-doc/sphinx/issues/6550>

Funzionalità aggiunte

- [#8022](#)¹⁷³⁵: autodoc: autodoc and autoattribute directives does not show right-hand value of the variable if docstring contains `:meta hide-value:` in info-field-list
- [#8514](#)¹⁷³⁶: autodoc: Default values of overloaded functions are taken from actual implementation if they're ellipsis
- [#8775](#)¹⁷³⁷: autodoc: Support type union operator (PEP-604) in Python 3.10 or above
- [#8297](#)¹⁷³⁸: autodoc: Allow to extend `autodoc_default_options` via directive options
- [#759](#)¹⁷³⁹: autodoc: Add a new configuration `autodoc_preserve_defaults` as an experimental feature. It preserves the default argument values of functions in source code and keep them not evaluated for readability.
- [#8619](#)¹⁷⁴⁰: html: kbd role generates customizable HTML tags for compound keys
- [#8634](#)¹⁷⁴¹: html: Allow to change the order of JS/CSS via `priority` parameter for `Sphinx.add_js_file()` and `Sphinx.add_css_file()`
- [#6241](#)¹⁷⁴²: html: Allow to add JS/CSS files to the specific page when an extension calls `app.add_js_file()` or `app.add_css_file()` on `html-page-context` event
- [#6550](#)¹⁷⁴³: html: Allow to use HTML permalink texts via `html_permaLinks_icon`
- [#1638](#)¹⁷⁴⁴: html: Add permalink icons to glossary terms
- [#8868](#)¹⁷⁴⁵: html search: performance issue with massive lists
- [#8867](#)¹⁷⁴⁶: html search: Update JavaScript stemmer code to the latest version of Snowball (v2.1.0)
- [#8852](#)¹⁷⁴⁷: i18n: Allow to translate heading syntax in MyST-Parser
- [#8649](#)¹⁷⁴⁸: imgconverter: Skip availability check if builder supports the image type
- [#8573](#)¹⁷⁴⁹: napoleon: Allow to change the style of custom sections using `napoleon_custom_sections`

¹⁷³⁵ <https://github.com/sphinx-doc/sphinx/issues/8022>

¹⁷³⁶ <https://github.com/sphinx-doc/sphinx/issues/8514>

¹⁷³⁷ <https://github.com/sphinx-doc/sphinx/issues/8775>

¹⁷³⁸ <https://github.com/sphinx-doc/sphinx/issues/8297>

¹⁷³⁹ <https://github.com/sphinx-doc/sphinx/issues/759>

¹⁷⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/8619>

¹⁷⁴¹ <https://github.com/sphinx-doc/sphinx/issues/8634>

¹⁷⁴² <https://github.com/sphinx-doc/sphinx/issues/6241>

¹⁷⁴³ <https://github.com/sphinx-doc/sphinx/issues/6550>

¹⁷⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/1638>

¹⁷⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/8868>

¹⁷⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/8867>

¹⁷⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/8852>

¹⁷⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/8649>

¹⁷⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/8573>

- #8004¹⁷⁵⁰: `napoleon`: Type definitions in Google style docstrings are rendered as references when `napoleon_preprocess_types` enabled
- #6241¹⁷⁵¹: `mathjax`: Include `mathjax.js` only on the document using equations
- #8775¹⁷⁵²: `py` domain: Support type union operator (PEP-604)
- #8651¹⁷⁵³: `std` domain: cross-reference for a rubric having inline item is broken
- #7642¹⁷⁵⁴: `std` domain: Optimize case-insensitive match of term
- #8681¹⁷⁵⁵: `viewcode`: Support incremental build
- #8132¹⁷⁵⁶: Add `project_copyright` as an alias of `copyright`
- #207¹⁷⁵⁷: Now `highlight_language` supports multiple languages
- #2030¹⁷⁵⁸: `code-block` and `literalinclude` supports automatic dedent via no-argument `:dedent:` option
- C++, also hyperlink operator overloads in expressions and alias declarations.
- #8247¹⁷⁵⁹: Allow production lists to refer to tokens from other production groups
- #8813¹⁷⁶⁰: Show what extension (or module) caused it on errors on event handler
- #8213¹⁷⁶¹: C++: add `maxdepth` option to `cpp:alias` to insert nested declarations.
- C, add `noroot` option to `c:alias` to render only nested declarations.
- C++, add `noroot` option to `cpp:alias` to render only nested declarations.

Bugs fixed

- #8727¹⁷⁶²: `apidoc`: namespace module file is not generated if no submodules there
- #741¹⁷⁶³: `autodoc`: `inherited-members` doesn't work for instance attributes on super class
- #8592¹⁷⁶⁴: `autodoc`: `:meta public:` does not effect to variables
- #8594¹⁷⁶⁵: `autodoc`: empty `__all__` attribute is ignored

¹⁷⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/8004>

¹⁷⁵¹ <https://github.com/sphinx-doc/sphinx/issues/6241>

¹⁷⁵² <https://github.com/sphinx-doc/sphinx/issues/8775>

¹⁷⁵³ <https://github.com/sphinx-doc/sphinx/issues/8651>

¹⁷⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/7642>

¹⁷⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/8681>

¹⁷⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/8132>

¹⁷⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/207>

¹⁷⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/2030>

¹⁷⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/8247>

¹⁷⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/8813>

¹⁷⁶¹ <https://github.com/sphinx-doc/sphinx/issues/8213>

¹⁷⁶² <https://github.com/sphinx-doc/sphinx/issues/8727>

¹⁷⁶³ <https://github.com/sphinx-doc/sphinx/issues/741>

¹⁷⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/8592>

¹⁷⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/8594>

- #8315¹⁷⁶⁶: autodoc: Failed to resolve struct.Struct type annotation
- #8652¹⁷⁶⁷: autodoc: All variable comments in the module are ignored if the module contains invalid type comments
- #8693¹⁷⁶⁸: autodoc: Default values for overloaded functions are rendered as string
- #8134¹⁷⁶⁹: autodoc: crashes when mocked decorator takes arguments
- #8800¹⁷⁷⁰: autodoc: Uninitialized attributes in superclass are recognized as undocumented
- #8655¹⁷⁷¹: autodoc: Failed to generate document if target module contains an object that raises an exception on `hasattr()`
- #8306¹⁷⁷²: autosummary: mocked modules are documented as empty page when using `:recursive:` option
- #8232¹⁷⁷³: graphviz: Image node is not rendered if graph file is in subdirectory
- #8618¹⁷⁷⁴: html: kbd role produces incorrect HTML when compound-key separators (`-`, `+` or `^`) are used as keystrokes
- #8629¹⁷⁷⁵: html: A type warning for `html_use_opensearch` is shown twice
- #8714¹⁷⁷⁶: html: kbd role with "Caps Lock" rendered incorrectly
- #8123¹⁷⁷⁷: html search: fix searching for terms containing `+` (Requires a custom search language that does not split on `+`)
- #8665¹⁷⁷⁸: html theme: Could not override `globaltoc_maxdepth` in `theme.conf`
- #8446¹⁷⁷⁹: html: consecutive spaces are displayed as single space
- #8745¹⁷⁸⁰: i18n: crashes with `KeyError` when translation message adds a new auto footnote reference
- #4304¹⁷⁸¹: linkcheck: Fix race condition that could lead to checking the availability of the same URL twice
- #8791¹⁷⁸²: linkcheck: The docname for each hyperlink is not displayed

¹⁷⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/8315>

¹⁷⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/8652>

¹⁷⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/8693>

¹⁷⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/8134>

¹⁷⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/8800>

¹⁷⁷¹ <https://github.com/sphinx-doc/sphinx/issues/8655>

¹⁷⁷² <https://github.com/sphinx-doc/sphinx/issues/8306>

¹⁷⁷³ <https://github.com/sphinx-doc/sphinx/issues/8232>

¹⁷⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/8618>

¹⁷⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/8629>

¹⁷⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/8714>

¹⁷⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/8123>

¹⁷⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/8665>

¹⁷⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/8446>

¹⁷⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/8745>

¹⁷⁸¹ <https://github.com/sphinx-doc/sphinx/issues/4304>

¹⁷⁸² <https://github.com/sphinx-doc/sphinx/issues/8791>

- #7118¹⁷⁸³: sphinx-quickstart: questionare got Mojibake if libreadline unavailable
- #8094¹⁷⁸⁴: texinfo: image files on the different directory with document are not copied
- #8782¹⁷⁸⁵: todo: Cross references in todolist get broken
- #8720¹⁷⁸⁶: viewcode: module pages are generated for epub on incremental build
- #8704¹⁷⁸⁷: viewcode: anchors are generated in incremental build after singlehtml
- #8756¹⁷⁸⁸: viewcode: highlighted code is generated even if not referenced
- #8671¹⁷⁸⁹: `highlight_options` is not working
- #8341¹⁷⁹⁰: C, fix intersphinx lookup types for names in declarations.
- C, C++: in general fix intersphinx and role lookup types.
- #8683¹⁷⁹¹: `html_last_updated_fmt` does not support UTC offset (%z)
- #8683¹⁷⁹²: `html_last_updated_fmt` generates wrong time zone for %Z
- #1112¹⁷⁹³: download role creates duplicated copies when relative path is specified
- #2616¹⁷⁹⁴ (fifth item): LaTeX: footnotes from captions are not clickable, and for manually numbered footnotes only first one with same number is an hyperlink
- #7576¹⁷⁹⁵: LaTeX with French babel and memoir crash: "Illegal parameter number in definition of \FNH@prefntext"
- #8055¹⁷⁹⁶: LaTeX (docs): A potential display bug with the LaTeX generation step in Sphinx (how to generate one-column index)
- #8072¹⁷⁹⁷: LaTeX: Directive `hlist` not implemented in LaTeX
- #8214¹⁷⁹⁸: LaTeX: The `index` role and the glossary generate duplicate entries in the LaTeX index (if both used for same term)
- #8735¹⁷⁹⁹: LaTeX: wrong internal links in pdf to captioned code-blocks when `numfig` is not True

¹⁷⁸³ <https://github.com/sphinx-doc/sphinx/issues/7118>

¹⁷⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/8094>

¹⁷⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/8782>

¹⁷⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/8720>

¹⁷⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/8704>

¹⁷⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/8756>

¹⁷⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/8671>

¹⁷⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/8341>

¹⁷⁹¹ <https://github.com/sphinx-doc/sphinx/issues/8683>

¹⁷⁹² <https://github.com/sphinx-doc/sphinx/issues/8683>

¹⁷⁹³ <https://github.com/sphinx-doc/sphinx/issues/1112>

¹⁷⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/2616>

¹⁷⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/7576>

¹⁷⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/8055>

¹⁷⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/8072>

¹⁷⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/8214>

¹⁷⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/8735>

- [#8442](#)¹⁸⁰⁰: LaTeX: some indexed terms are ignored when using xelatex engine (or pdflatex and `latex_use_xindy` set to True) with memoir class
- [#8750](#)¹⁸⁰¹: LaTeX: URLs as footnotes fail to show in PDF if originating from inside function type signatures
- [#8780](#)¹⁸⁰²: LaTeX: long words in narrow columns may not be hyphenated
- [#8788](#)¹⁸⁰³: LaTeX: `\titleformat` last argument in sphinx.sty should be bracketed, not braced (and is anyhow not needed)
- [#8849](#)¹⁸⁰⁴: LaTeX: code-block printed out of margin (see the opt-in LaTeX syntax boolean `verbatimforcewraps` for use via the '`sphinxsetup`' key of `latex_elements`)
- [#8183](#)¹⁸⁰⁵: LaTeX: Remove `substitution_reference` nodes from doctree only on LaTeX builds
- [#8865](#)¹⁸⁰⁶: LaTeX: Restructure the index nodes inside title nodes only on LaTeX builds
- [#8796](#)¹⁸⁰⁷: LaTeX: potentially critical low level TeX coding mistake has gone unnoticed so far
- C, `c:alias` skip symbols without explicit declarations instead of crashing.
- C, `c:alias` give a warning when the root symbol is not declared.
- C, `expr` role should start symbol lookup in the current scope.

14.60 Release 3.4.3 (released Jan 08, 2021)

Bugs fixed

- [#8655](#)¹⁸⁰⁸: autodoc: Failed to generate document if target module contains an object that raises an exception on `hasattr()`

¹⁸⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/8442>

¹⁸⁰¹ <https://github.com/sphinx-doc/sphinx/issues/8750>

¹⁸⁰² <https://github.com/sphinx-doc/sphinx/issues/8780>

¹⁸⁰³ <https://github.com/sphinx-doc/sphinx/issues/8788>

¹⁸⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/8849>

¹⁸⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/8183>

¹⁸⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/8865>

¹⁸⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/8796>

¹⁸⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/8655>

14.61 Release 3.4.2 (released Jan 04, 2021)

Bugs fixed

- [#8164](#)¹⁸⁰⁹: autodoc: Classes that inherit mocked class are not documented
- [#8602](#)¹⁸¹⁰: autodoc: The autodoc-process-docstring event is emitted to the non-datadescriptors unexpectedly
- [#8616](#)¹⁸¹¹: autodoc: AttributeError is raised on non-class object is passed to autoclass directive

14.62 Release 3.4.1 (released Dec 25, 2020)

Bugs fixed

- [#8559](#)¹⁸¹²: autodoc: AttributeError is raised when using forward-reference type annotations
- [#8568](#)¹⁸¹³: autodoc: TypeError is raised on checking slots attribute
- [#8567](#)¹⁸¹⁴: autodoc: Instance attributes are incorrectly added to Parent class
- [#8566](#)¹⁸¹⁵: autodoc: The autodoc-process-docstring event is emitted to the alias classes unexpectedly
- [#8583](#)¹⁸¹⁶: autodoc: Unnecessary object comparison via `__eq__` method
- [#8565](#)¹⁸¹⁷: linkcheck: Fix PriorityQueue crash when link tuples are not comparable

14.63 Release 3.4.0 (released Dec 20, 2020)

Modifiche incompatibili

- [#8105](#)¹⁸¹⁸: autodoc: the signature of class constructor will be shown for decorated classes, not a signature of decorator

¹⁸⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/8164>

¹⁸¹⁰ <https://github.com/sphinx-doc/sphinx/issues/8602>

¹⁸¹¹ <https://github.com/sphinx-doc/sphinx/issues/8616>

¹⁸¹² <https://github.com/sphinx-doc/sphinx/issues/8559>

¹⁸¹³ <https://github.com/sphinx-doc/sphinx/issues/8568>

¹⁸¹⁴ <https://github.com/sphinx-doc/sphinx/issues/8567>

¹⁸¹⁵ <https://github.com/sphinx-doc/sphinx/issues/8566>

¹⁸¹⁶ <https://github.com/sphinx-doc/sphinx/issues/8583>

¹⁸¹⁷ <https://github.com/sphinx-doc/sphinx/issues/8565>

¹⁸¹⁸ <https://github.com/sphinx-doc/sphinx/issues/8105>

Deprecato

- L'argomento `follow_wrapped` di `sphinx.util.inspect.signature()`
- L'argomento `no_docstring` di `sphinx.ext.autodoc.Documenter.add_content()`
- `sphinx.ext.autodoc.Documenter.get_object_members()`
- `sphinx.ext.autodoc.DataDeclarationDocumenter`
- `sphinx.ext.autodoc.GenericAliasDocumenter`
- `sphinx.ext.autodoc.InstanceAttributeDocumenter`
- `sphinx.ext.autodoc.SlotsAttributeDocumenter`
- `sphinx.ext.autodoc.TypeVarDocumenter`
- `sphinx.ext.autodoc.importer._getannotations()`
- `sphinx.ext.autodoc.importer._getmro()`
- `sphinx.pycode.ModuleAnalyzer.parse()`
- `sphinx.util.osutil.movefile()`
- `sphinx.util.requests.is_ssl_error()`

Funzionalità aggiunte

- [#8119](#)¹⁸¹⁹: autodoc: Allow to determine whether a member not included in `__all__` attribute of the module should be documented or not via `autodoc-skip-member` event
- [#8219](#)¹⁸²⁰: autodoc: Parameters for generic class are not shown when super class is a generic class and show-inheritance option is given (in Python 3.7 or above)
- autodoc: Add `Documenter.config` as a shortcut to access the config object
- autodoc: Add `Optional[t]` to annotation of function and method if a default value equal to `None` is set.
- [#8209](#)¹⁸²¹: autodoc: Add `:no-value:` option to `autoattribute` and `autodata` directive to suppress the default value of the variable
- [#8460](#)¹⁸²²: autodoc: Support custom types defined by `typing.NewType`
- [#8285](#)¹⁸²³: napoleon: Add `napoleon_attr_annotations` to merge type hints on source code automatically if any type is specified in docstring
- [#8236](#)¹⁸²⁴: napoleon: Support numpydoc's "Receives" section

¹⁸¹⁹ <https://github.com/sphinx-doc/sphinx/issues/8119>

¹⁸²⁰ <https://github.com/sphinx-doc/sphinx/issues/8219>

¹⁸²¹ <https://github.com/sphinx-doc/sphinx/issues/8209>

¹⁸²² <https://github.com/sphinx-doc/sphinx/issues/8460>

¹⁸²³ <https://github.com/sphinx-doc/sphinx/issues/8285>

¹⁸²⁴ <https://github.com/sphinx-doc/sphinx/issues/8236>

- [#6914](#)¹⁸²⁵: Add a new event *warn-missing-reference* to custom warning messages when failed to resolve a cross-reference
- [#6914](#)¹⁸²⁶: Emit a detailed warning when failed to resolve a `:ref:` reference
- [#6629](#)¹⁸²⁷: `linkcheck:` The builder now handles rate limits. See *linkcheck_rate_limit_timeout* for details.

Bugs fixed

- [#7613](#)¹⁸²⁸: autodoc: autodoc does not respect `__signature__` of the class
- [#4606](#)¹⁸²⁹: autodoc: the location of the warning is incorrect for inherited method
- [#8105](#)¹⁸³⁰: autodoc: the signature of class constructor is incorrect if the class is decorated
- [#8434](#)¹⁸³¹: autodoc: *autodoc_type_aliases* does not effect to variables and attributes
- [#8443](#)¹⁸³²: autodoc: autodata directive can't create document for PEP-526 based type annotated variables
- [#8443](#)¹⁸³³: autodoc: autoattribute directive can't create document for PEP-526 based uninitialized variables
- [#8480](#)¹⁸³⁴: autodoc: autoattribute could not create document for `__slots__` attributes
- [#8503](#)¹⁸³⁵: autodoc: autoattribute could not create document for a GenericAlias as class attributes correctly
- [#8534](#)¹⁸³⁶: autodoc: autoattribute could not create document for a commented attribute in alias class
- [#8452](#)¹⁸³⁷: autodoc: *autodoc_type_aliases* doesn't work when *autodoc_typehints* is set to "description"
- [#8541](#)¹⁸³⁸: autodoc: *autodoc_type_aliases* doesn't work for the type annotation to instance attributes
- [#8460](#)¹⁸³⁹: autodoc: autodata and autoattribute directives do not display type information of TypeVars

¹⁸²⁵ <https://github.com/sphinx-doc/sphinx/issues/6914>

¹⁸²⁶ <https://github.com/sphinx-doc/sphinx/issues/6914>

¹⁸²⁷ <https://github.com/sphinx-doc/sphinx/issues/6629>

¹⁸²⁸ <https://github.com/sphinx-doc/sphinx/issues/7613>

¹⁸²⁹ <https://github.com/sphinx-doc/sphinx/issues/4606>

¹⁸³⁰ <https://github.com/sphinx-doc/sphinx/issues/8105>

¹⁸³¹ <https://github.com/sphinx-doc/sphinx/issues/8434>

¹⁸³² <https://github.com/sphinx-doc/sphinx/issues/8443>

¹⁸³³ <https://github.com/sphinx-doc/sphinx/issues/8443>

¹⁸³⁴ <https://github.com/sphinx-doc/sphinx/issues/8480>

¹⁸³⁵ <https://github.com/sphinx-doc/sphinx/issues/8503>

¹⁸³⁶ <https://github.com/sphinx-doc/sphinx/issues/8534>

¹⁸³⁷ <https://github.com/sphinx-doc/sphinx/issues/8452>

¹⁸³⁸ <https://github.com/sphinx-doc/sphinx/issues/8541>

¹⁸³⁹ <https://github.com/sphinx-doc/sphinx/issues/8460>

- #8493¹⁸⁴⁰: autodoc: references to builtins not working in class aliases
- #8522¹⁸⁴¹: autodoc: `__bool__` method could be called
- #8067¹⁸⁴²: autodoc: A typehint for the instance variable having `type_comment` on super class is not displayed
- #8545¹⁸⁴³: autodoc: a `__slots__` attribute is not documented even having docstring
- #741¹⁸⁴⁴: autodoc: `inherited-members` doesn't work for instance attributes on super class
- #8477¹⁸⁴⁵: autosummary: non utf-8 reST files are generated when template contains multibyte characters
- #8501¹⁸⁴⁶: autosummary: summary extraction splits text after "el at." unexpectedly
- #8524¹⁸⁴⁷: html: Wrong `url_root` has been generated on a document named "index"
- #8419¹⁸⁴⁸: html search: Do not load `language_data.js` in non-search pages
- #8549¹⁸⁴⁹: i18n: `-D gettext_compact=0` is no longer working
- #8454¹⁸⁵⁰: graphviz: The layout option for graph and digraph directives don't work
- #8131¹⁸⁵¹: linkcheck: Use GET when HEAD requests cause Too Many Redirects, to accommodate infinite redirect loops on HEAD
- #8437¹⁸⁵²: Makefile: `make clean` with empty `BUILDDIR` is dangerous
- #8365¹⁸⁵³: py domain: `:type:` and `:rtype:` gives false ambiguous class lookup warnings
- #8352¹⁸⁵⁴: std domain: Failed to parse an option that starts with bracket
- #8519¹⁸⁵⁵: LaTeX: Prevent page brake in the middle of a `seealso`
- #8520¹⁸⁵⁶: C, fix copying of `AliasNode`.

¹⁸⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/8493>

¹⁸⁴¹ <https://github.com/sphinx-doc/sphinx/issues/8522>

¹⁸⁴² <https://github.com/sphinx-doc/sphinx/issues/8067>

¹⁸⁴³ <https://github.com/sphinx-doc/sphinx/issues/8545>

¹⁸⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/741>

¹⁸⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/8477>

¹⁸⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/8501>

¹⁸⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/8524>

¹⁸⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/8419>

¹⁸⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/8549>

¹⁸⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/8454>

¹⁸⁵¹ <https://github.com/sphinx-doc/sphinx/issues/8131>

¹⁸⁵² <https://github.com/sphinx-doc/sphinx/issues/8437>

¹⁸⁵³ <https://github.com/sphinx-doc/sphinx/issues/8365>

¹⁸⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/8352>

¹⁸⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/8519>

¹⁸⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/8520>

14.64 Release 3.3.1 (released Nov 12, 2020)

Bugs fixed

- [#8372](#)¹⁸⁵⁷: autodoc: autoclass directive became slower than Sphinx 3.2
- [#7727](#)¹⁸⁵⁸: autosummary: raise PycodeError when documenting python package without `__init__.py`
- [#8350](#)¹⁸⁵⁹: autosummary: autosummary_mock_imports causes slow down builds
- [#8364](#)¹⁸⁶⁰: C, properly initialize attributes in empty symbols.
- [#8399](#)¹⁸⁶¹: i18n: Put system locale path after the paths specified by configuration

14.65 Release 3.3.0 (released Nov 02, 2020)

Deprecato

- `sphinx.builders.latex.LaTeXBuilder.usepackages`
- `sphinx.builders.latex.LaTeXBuilder.usepackages_afger_hyperref`
- `sphinx.ext.autodoc.SingledispatchFunctionDocumenter`
- `sphinx.ext.autodoc.SingledispatchMethodDocumenter`

Funzionalità aggiunte

- [#8100](#)¹⁸⁶²: html: Show a better error message for failures on copying `html_static_files`
- [#8141](#)¹⁸⁶³: C: added a `maxdepth` option to `c:alias` to insert nested declarations.
- [#8081](#)¹⁸⁶⁴: LaTeX: Allow to add LaTeX package via `app.add_latex_package()` until just before writing `.tex` file
- [#7996](#)¹⁸⁶⁵: manpage: Add `man_make_section_directory` to make a section directory on build man page
- [#8289](#)¹⁸⁶⁶: epub: Allow to suppress "duplicated ToC entry found" warnings from epub builder using `suppress_warnings`.

¹⁸⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/8372>

¹⁸⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/7727>

¹⁸⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/8350>

¹⁸⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/8364>

¹⁸⁶¹ <https://github.com/sphinx-doc/sphinx/issues/8399>

¹⁸⁶² <https://github.com/sphinx-doc/sphinx/issues/8100>

¹⁸⁶³ <https://github.com/sphinx-doc/sphinx/issues/8141>

¹⁸⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/8081>

¹⁸⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/7996>

¹⁸⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/8289>

- #8298¹⁸⁶⁷: sphinx-quickstart: Add `sphinx-quickstart --no-sep` option
- #8304¹⁸⁶⁸: sphinx.testing: Register public markers in sphinx.testing.fixtures
- #8051¹⁸⁶⁹: napoleon: use the obj role for all See Also items
- #8050¹⁸⁷⁰: napoleon: Apply `napoleon_preprocess_types` to every field
- C and C++, show line numbers for previous declarations when duplicates are detected.
- #8183¹⁸⁷¹: Remove substitution_reference nodes from doctree only on LaTeX builds

Bugs fixed

- #8085¹⁸⁷²: i18n: Add support for having single text domain
- #6640¹⁸⁷³: i18n: Failed to override system message translation
- #8143¹⁸⁷⁴: autodoc: `AttributeError` is raised when `False` value is passed to `autodoc_default_options`
- #8103¹⁸⁷⁵: autodoc: `functools.cached_property` is not considered as a property
- #8190¹⁸⁷⁶: autodoc: parsing error is raised if some extension replaces docstring by string not ending with blank lines
- #8142¹⁸⁷⁷: autodoc: Wrong constructor signature for the class derived from `typing.Generic`
- #8157¹⁸⁷⁸: autodoc: `TypeError` is raised when annotation has invalid `__args__`
- #7964¹⁸⁷⁹: autodoc: Tuple in default value is wrongly rendered
- #8200¹⁸⁸⁰: autodoc: type aliases break type formatting of `autoattribute`
- #7786¹⁸⁸¹: autodoc: can't detect overloaded methods defined in other file
- #8294¹⁸⁸²: autodoc: single-string `__slots__` is not handled correctly
- #7785¹⁸⁸³: autodoc: `autodoc_typehints='none'` does not effect to overloaded functions

¹⁸⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/8298>

¹⁸⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/8304>

¹⁸⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/8051>

¹⁸⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/8050>

¹⁸⁷¹ <https://github.com/sphinx-doc/sphinx/issues/8183>

¹⁸⁷² <https://github.com/sphinx-doc/sphinx/issues/8085>

¹⁸⁷³ <https://github.com/sphinx-doc/sphinx/issues/6640>

¹⁸⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/8143>

¹⁸⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/8103>

¹⁸⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/8190>

¹⁸⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/8142>

¹⁸⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/8157>

¹⁸⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/7964>

¹⁸⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/8200>

¹⁸⁸¹ <https://github.com/sphinx-doc/sphinx/issues/7786>

¹⁸⁸² <https://github.com/sphinx-doc/sphinx/issues/8294>

¹⁸⁸³ <https://github.com/sphinx-doc/sphinx/issues/7785>

- [#8192¹⁸⁸⁴](#): napoleon: description is disappeared when it contains inline literals
- [#8142¹⁸⁸⁵](#): napoleon: Potential of regex denial of service in google style docs
- [#8169¹⁸⁸⁶](#): LaTeX: pxjahyper loaded even when latex_engine is not platex
- [#8215¹⁸⁸⁷](#): LaTeX: 'oneside' classoption causes build warning
- [#8175¹⁸⁸⁸](#): intersphinx: Potential of regex denial of service by broken inventory
- [#8277¹⁸⁸⁹](#): sphinx-build: missing and redundant spacing (and etc) for console output on building
- [#7973¹⁸⁹⁰](#): imgconverter: Check availability of imagemagick many times
- [#8255¹⁸⁹¹](#): py domain: number in default argument value is changed from hexadecimal to decimal
- [#8316¹⁸⁹²](#): html: Prevent arrow keys changing page when button elements are focused
- [#8343¹⁸⁹³](#): html search: Fix unnecessary load of images when parsing the document
- [#8254¹⁸⁹⁴](#): html theme: Line numbers misalign with code lines
- [#8093¹⁸⁹⁵](#): The highlight warning has wrong location in some builders (LaTeX, singlehtml and so on)
- [#8215¹⁸⁹⁶](#): Eliminate Fancyhdr build warnings for oneside documents
- [#8239¹⁸⁹⁷](#): Failed to refer a token in productionlist if it is indented
- [#8268¹⁸⁹⁸](#): linkcheck: Report HTTP errors when linkcheck_anchors is True
- [#8245¹⁸⁹⁹](#): linkcheck: take source directory into account for local files
- [#8321¹⁹⁰⁰](#): linkcheck: tel: schema hyperlinks are detected as errors
- [#8323¹⁹⁰¹](#): linkcheck: An exit status is incorrect when links having unsupported schema found
- [#8188¹⁹⁰²](#): C, add missing items to internal object types dictionary, e.g., preventing

¹⁸⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/8192>

¹⁸⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/8142>

¹⁸⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/8169>

¹⁸⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/8215>

¹⁸⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/8175>

¹⁸⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/8277>

¹⁸⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/7973>

¹⁸⁹¹ <https://github.com/sphinx-doc/sphinx/issues/8255>

¹⁸⁹² <https://github.com/sphinx-doc/sphinx/issues/8316>

¹⁸⁹³ <https://github.com/sphinx-doc/sphinx/issues/8343>

¹⁸⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/8254>

¹⁸⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/8093>

¹⁸⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/8215>

¹⁸⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/8239>

¹⁸⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/8268>

¹⁸⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/8245>

¹⁹⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/8321>

¹⁹⁰¹ <https://github.com/sphinx-doc/sphinx/issues/8323>

¹⁹⁰² <https://github.com/sphinx-doc/sphinx/issues/8188>

intersphinx from resolving them.

- C, fix anon objects in intersphinx.
- #8270¹⁹⁰³, C++, properly reject functions as duplicate declarations if a non-function declaration of the same name already exists.
- C, fix references to function parameters. Link to the function instead of a non-existing anchor.
- #6914¹⁹⁰⁴: figure numbers are unexpectedly assigned to uncaptioned items
- #8320¹⁹⁰⁵: make "inline" line numbers un-selectable

Testing

- #8257¹⁹⁰⁶: Support parallel build in sphinx.testing

14.66 Release 3.2.1 (released Aug 14, 2020)

Funzionalità aggiunte

- #8095¹⁹⁰⁷: `napoleon`: Add `napoleon_preprocess_types` to enable the type preprocessor for numpy style docstrings
- #8114¹⁹⁰⁸: C and C++, parse function attributes after parameters and qualifiers.

Bugs fixed

- #8074¹⁹⁰⁹: `napoleon`: Crashes during processing C-ext module
- #8088¹⁹¹⁰: `napoleon`: "Inline literal start-string without end-string" warning in Numpy style Parameters section
- #8084¹⁹¹¹: `autodoc`: `KeyError` is raised on documenting an attribute of the broken class
- #8091¹⁹¹²: `autodoc`: `AttributeError` is raised on documenting an attribute on Python 3.5.2
- #8099¹⁹¹³: `autodoc`: `NameError` is raised when target code uses `TYPE_CHECKING`

¹⁹⁰³ <https://github.com/sphinx-doc/sphinx/issues/8270>

¹⁹⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/6914>

¹⁹⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/8320>

¹⁹⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/8257>

¹⁹⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/8095>

¹⁹⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/8114>

¹⁹⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/8074>

¹⁹¹⁰ <https://github.com/sphinx-doc/sphinx/issues/8088>

¹⁹¹¹ <https://github.com/sphinx-doc/sphinx/issues/8084>

¹⁹¹² <https://github.com/sphinx-doc/sphinx/issues/8091>

¹⁹¹³ <https://github.com/sphinx-doc/sphinx/issues/8099>

- C++, fix parsing of template template parameters, broken by the fix of #7944¹⁹¹⁴

14.67 Release 3.2.0 (released Aug 08, 2020)

Deprecato

- `sphinx.ext.autodoc.members_set_option()`
- `sphinx.ext.autodoc.merge_special_members_option()`
- `sphinx.writers.texinfo.TexinfoWriter.desc`
- C, parsing of pre-v3 style type directives and roles, along with the options `c_allow_pre_v3` and `c_warn_on_allowed_pre_v3`.

Funzionalità aggiunte

- #2076¹⁹¹⁵: autodoc: Allow overriding of exclude-members in skip-member function
- #8034¹⁹¹⁶: autodoc: `:private-member:` can take an explicit list of member names to be documented
- #2024¹⁹¹⁷: autosummary: Add `autosummary_filename_map` to avoid conflict of filenames between two object with different case
- #8011¹⁹¹⁸: autosummary: Support instance attributes as a target of autosummary directive
- #7849¹⁹¹⁹: html: Add `html_codeblock_linenos_style` to change the style of line numbers for code-blocks
- #7853¹⁹²⁰: C and C++, support parameterized GNU style attributes.
- #7888¹⁹²¹: napoleon: Add aliases Warn and Raise.
- #7690¹⁹²²: napoleon: parse type strings and make them hyperlinks as possible. The conversion rule can be updated via `napoleon_type_aliases`
- #8049¹⁹²³: napoleon: Create a hyperlink for each the type of parameter when `napoleon_use_param` is False
- C, added `c:alias` directive for inserting copies of existing declarations.

¹⁹¹⁴ <https://github.com/sphinx-doc/sphinx/issues/7944>

¹⁹¹⁵ <https://github.com/sphinx-doc/sphinx/issues/2076>

¹⁹¹⁶ <https://github.com/sphinx-doc/sphinx/issues/8034>

¹⁹¹⁷ <https://github.com/sphinx-doc/sphinx/issues/2024>

¹⁹¹⁸ <https://github.com/sphinx-doc/sphinx/issues/8011>

¹⁹¹⁹ <https://github.com/sphinx-doc/sphinx/issues/7849>

¹⁹²⁰ <https://github.com/sphinx-doc/sphinx/issues/7853>

¹⁹²¹ <https://github.com/sphinx-doc/sphinx/issues/7888>

¹⁹²² <https://github.com/sphinx-doc/sphinx/issues/7690>

¹⁹²³ <https://github.com/sphinx-doc/sphinx/issues/8049>

- #7745¹⁹²⁴: html: inventory is broken if the docname contains a space
- #7991¹⁹²⁵: html search: Allow searching for numbers
- #7902¹⁹²⁶: html theme: Add a new option `globaltoc_maxdepth` to control the behavior of `globaltoc` in sidebar
- #7840¹⁹²⁷: i18n: Optimize the dependencies check on bootstrap
- #7768¹⁹²⁸: i18n: `figure_language_filename` supports `docpath` token
- #5208¹⁹²⁹: linkcheck: Support checks for local links
- #5090¹⁹³⁰: setuptools: Link verbosity to `distutils`' `-v` and `-q` option
- #6698¹⁹³¹: doctest: Add `:trim-doctest-flags:` and `:no-trim-doctest-flags:` options to doctest, testcode and testoutput directives
- #7052¹⁹³²: add `:noindexentry:` to the Python, C, C++, and Javascript domains. Update the documentation to better reflect the relationship between this option and the `:noindex:` option.
- #7899¹⁹³³: C, add possibility of parsing of some pre-v3 style type directives and roles and try to convert them to equivalent v3 directives/roles. Set the new option `c_allow_pre_v3` to `True` to enable this. The warnings printed from this functionality can be suppressed by setting `c_warn_on_allowed_pre_v3` to `True`. The functionality is immediately deprecated.
- #7999¹⁹³⁴: C, add support for named variadic macro arguments.
- #8071¹⁹³⁵: Allow to suppress "self referenced toctrees" warning

Bugs fixed

- #7886¹⁹³⁶: autodoc: `TypeError` is raised on mocking generic-typed classes
- #7935¹⁹³⁷: autodoc: function signature is not shown when the function has a parameter having `inspect._empty` as its default value
- #7901¹⁹³⁸: autodoc: type annotations for overloaded functions are not resolved

¹⁹²⁴ <https://github.com/sphinx-doc/sphinx/issues/7745>

¹⁹²⁵ <https://github.com/sphinx-doc/sphinx/issues/7991>

¹⁹²⁶ <https://github.com/sphinx-doc/sphinx/issues/7902>

¹⁹²⁷ <https://github.com/sphinx-doc/sphinx/issues/7840>

¹⁹²⁸ <https://github.com/sphinx-doc/sphinx/issues/7768>

¹⁹²⁹ <https://github.com/sphinx-doc/sphinx/issues/5208>

¹⁹³⁰ <https://github.com/sphinx-doc/sphinx/issues/5090>

¹⁹³¹ <https://github.com/sphinx-doc/sphinx/issues/6698>

¹⁹³² <https://github.com/sphinx-doc/sphinx/issues/7052>

¹⁹³³ <https://github.com/sphinx-doc/sphinx/issues/7899>

¹⁹³⁴ <https://github.com/sphinx-doc/sphinx/issues/7999>

¹⁹³⁵ <https://github.com/sphinx-doc/sphinx/issues/8071>

¹⁹³⁶ <https://github.com/sphinx-doc/sphinx/issues/7886>

¹⁹³⁷ <https://github.com/sphinx-doc/sphinx/issues/7935>

¹⁹³⁸ <https://github.com/sphinx-doc/sphinx/issues/7901>

- [#904](#)¹⁹³⁹: autodoc: An instance attribute cause a crash of autofunction directive
- [#1362](#)¹⁹⁴⁰: autodoc: `private-members` option does not work for class attributes
- [#7983](#)¹⁹⁴¹: autodoc: Generator type annotation is wrongly rendered in py36
- [#8030](#)¹⁹⁴²: autodoc: An uninitialized annotated instance variable is not documented when `:inherited-members:` option given
- [#8032](#)¹⁹⁴³: autodoc: A type hint for the instance variable defined at parent class is not shown in the document of the derived class
- [#8041](#)¹⁹⁴⁴: autodoc: An annotated instance variable on super class is not documented when derived class has other annotated instance variables
- [#7839](#)¹⁹⁴⁵: autosummary: cannot handle umlauts in function names
- [#7865](#)¹⁹⁴⁶: autosummary: Failed to extract summary line when abbreviations found
- [#7866](#)¹⁹⁴⁷: autosummary: Failed to extract correct summary line when docstring contains a hyperlink target
- [#7469](#)¹⁹⁴⁸: autosummary: "Module attributes" header is not translatable
- [#7940](#)¹⁹⁴⁹: apidoc: An extra newline is generated at the end of the rst file if a module has submodules
- [#4258](#)¹⁹⁵⁰: napoleon: decorated special methods are not shown
- [#7799](#)¹⁹⁵¹: napoleon: parameters are not escaped for combined params in numpydoc
- [#7780](#)¹⁹⁵²: napoleon: multiple parameters declaration in numpydoc was wrongly recognized when `napoleon_use_param=True`
- [#7715](#)¹⁹⁵³: LaTeX: `numfig_secnum_depth > 1` leads to wrong figure links
- [#7846](#)¹⁹⁵⁴: html theme: XML-invalid files were generated
- [#7894](#)¹⁹⁵⁵: gettext: Wrong source info is shown when using `rst_epilog`
- [#7691](#)¹⁹⁵⁶: linkcheck: HEAD requests are not used for checking

¹⁹³⁹ <https://github.com/sphinx-doc/sphinx/issues/904>

¹⁹⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/1362>

¹⁹⁴¹ <https://github.com/sphinx-doc/sphinx/issues/7983>

¹⁹⁴² <https://github.com/sphinx-doc/sphinx/issues/8030>

¹⁹⁴³ <https://github.com/sphinx-doc/sphinx/issues/8032>

¹⁹⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/8041>

¹⁹⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/7839>

¹⁹⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/7865>

¹⁹⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/7866>

¹⁹⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/7469>

¹⁹⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/7940>

¹⁹⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/4258>

¹⁹⁵¹ <https://github.com/sphinx-doc/sphinx/issues/7799>

¹⁹⁵² <https://github.com/sphinx-doc/sphinx/issues/7780>

¹⁹⁵³ <https://github.com/sphinx-doc/sphinx/issues/7715>

¹⁹⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/7846>

¹⁹⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/7894>

¹⁹⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/7691>

- #4888¹⁹⁵⁷: i18n: Failed to add an explicit title to `:ref:` role on translation
- #7928¹⁹⁵⁸: py domain: failed to resolve a type annotation for the attribute
- #8008¹⁹⁵⁹: py domain: failed to parse a type annotation containing ellipsis
- #7994¹⁹⁶⁰: std domain: option directive does not generate old `node_id` compatible with 2.x or older
- #7968¹⁹⁶¹: i18n: The content of `math` directive is interpreted as reST on translation
- #7768¹⁹⁶²: i18n: The root element for `figure_language_filename` is not a path that user specifies in the document
- #7993¹⁹⁶³: texinfo: `TypeError` is raised for nested object descriptions
- #7993¹⁹⁶⁴: texinfo: a warning not supporting `desc_signature_line` node is shown
- #7869¹⁹⁶⁵: `abbr` role without an explanation will show the explanation from the previous `abbr` role
- #8048¹⁹⁶⁶: graphviz: `graphviz.css` was copied on building non-HTML document
- C and C++, removed `noindex` directive option as it did nothing.
- #7619¹⁹⁶⁷: Duplicated node IDs are generated if node has multiple IDs
- #2050¹⁹⁶⁸: Symbols sections are appeared twice in the index page
- #8017¹⁹⁶⁹: Fix circular import in `sphinx.addnodes`
- #7986¹⁹⁷⁰: CSS: make "highlight" selector more robust
- #7944¹⁹⁷¹: C++, parse non-type template parameters starting with a dependent qualified name.
- C, don't deepcopy the entire symbol table and make a mess every time an enumerator is handled.

¹⁹⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/4888>

¹⁹⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/7928>

¹⁹⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/8008>

¹⁹⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/7994>

¹⁹⁶¹ <https://github.com/sphinx-doc/sphinx/issues/7968>

¹⁹⁶² <https://github.com/sphinx-doc/sphinx/issues/7768>

¹⁹⁶³ <https://github.com/sphinx-doc/sphinx/issues/7993>

¹⁹⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/7993>

¹⁹⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/7869>

¹⁹⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/8048>

¹⁹⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/7619>

¹⁹⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/2050>

¹⁹⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/8017>

¹⁹⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/7986>

¹⁹⁷¹ <https://github.com/sphinx-doc/sphinx/issues/7944>

14.68 Release 3.1.2 (released Jul 05, 2020)

Modifiche incompatibili

- #7650¹⁹⁷²: autodoc: the signature of base function will be shown for decorated functions, not a signature of decorator

Bugs fixed

- #7844¹⁹⁷³: autodoc: Failed to detect module when relative module name given
- #7856¹⁹⁷⁴: autodoc: AttributeError is raised when non-class object is given to the autoclass directive
- #7850¹⁹⁷⁵: autodoc: KeyError is raised for invalid mark up when autodoc_typehints is 'description'
- #7812¹⁹⁷⁶: autodoc: crashed if the target name matches to both an attribute and module that are same name
- #7650¹⁹⁷⁷: autodoc: function signature becomes (*args, **kwargs) if the function is decorated by generic decorator
- #7812¹⁹⁷⁸: autosummary: generates broken stub files if the target code contains an attribute and module that are same name
- #7806¹⁹⁷⁹: viewcode: Failed to resolve viewcode references on 3rd party builders
- #7838¹⁹⁸⁰: html theme: List items have extra vertical space
- #7878¹⁹⁸¹: html theme: Undesired interaction between "overflow" and "float"

¹⁹⁷² <https://github.com/sphinx-doc/sphinx/issues/7650>

¹⁹⁷³ <https://github.com/sphinx-doc/sphinx/issues/7844>

¹⁹⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/7856>

¹⁹⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/7850>

¹⁹⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/7812>

¹⁹⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/7650>

¹⁹⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/7812>

¹⁹⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/7806>

¹⁹⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/7838>

¹⁹⁸¹ <https://github.com/sphinx-doc/sphinx/issues/7878>

14.69 Release 3.1.1 (released Jun 14, 2020)

Modifiche incompatibili

- #7808¹⁹⁸²: napoleon: a type for attribute are represented as typed field

Funzionalità aggiunte

- #7807¹⁹⁸³: autodoc: Show detailed warning when type_comment is mismatched with its signature

Bugs fixed

- #7808¹⁹⁸⁴: autodoc: Warnings raised on variable and attribute type annotations
- #7802¹⁹⁸⁵: autodoc: EOFError is raised on parallel build
- #7821¹⁹⁸⁶: autodoc: TypeError is raised for overloaded C-ext function
- #7805¹⁹⁸⁷: autodoc: an object which descriptors returns is unexpectedly documented
- #7807¹⁹⁸⁸: autodoc: wrong signature is shown for the function using contextmanager
- #7812¹⁹⁸⁹: autosummary: generates broken stub files if the target code contains an attribute and module that are same name
- #7808¹⁹⁹⁰: napoleon: Warnings raised on variable and attribute type annotations
- #7811¹⁹⁹¹: sphinx.util.inspect causes circular import problem

14.70 Release 3.1.0 (released Jun 08, 2020)

Dipendenze

- #7746¹⁹⁹²: mathjax: Update to 2.7.5

¹⁹⁸² <https://github.com/sphinx-doc/sphinx/issues/7808>

¹⁹⁸³ <https://github.com/sphinx-doc/sphinx/issues/7807>

¹⁹⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/7808>

¹⁹⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/7802>

¹⁹⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/7821>

¹⁹⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/7805>

¹⁹⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/7807>

¹⁹⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/7812>

¹⁹⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/7808>

¹⁹⁹¹ <https://github.com/sphinx-doc/sphinx/issues/7811>

¹⁹⁹² <https://github.com/sphinx-doc/sphinx/issues/7746>

Modifiche incompatibili

- [#7477](#)¹⁹⁹³: `imgconverter`: Invoke "magick convert" command by default on Windows

Deprecato

- The first argument for `sphinx.ext.autosummary.generate.AutosummaryRenderer` has been changed to Sphinx object
- `sphinx.ext.autosummary.generate.AutosummaryRenderer` accetta un tipo di oggetto come argomento
- L'argomento `ignore` di `sphinx.ext.autodoc.Documenter.get_doc()`
- The `template_dir` argument of `sphinx.ext.autosummary.generate.AutosummaryRenderer`
- The `module` argument of `sphinx.ext.autosummary.generate.find_autosummary_in_docstring()`
- The `builder` argument of `sphinx.ext.autosummary.generate.generate_autosummary_docs()`
- The `template_dir` argument of `sphinx.ext.autosummary.generate.generate_autosummary_docs()`
- L'argomento `ignore` di `sphinx.util.docstring.prepare_docstring()`
- `sphinx.ext.autosummary.generate.AutosummaryRenderer.exists()`
- `sphinx.util.rpartition()`

Funzionalità aggiunte

- LaTeX: Make the `toplevel_sectioning` setting optional in LaTeX theme
- LaTeX: Allow to override `papersize` and `pointsize` from LaTeX themes
- LaTeX: Add `latex_theme_options` to override theme options
- [#7410](#)¹⁹⁹⁴: Allow to suppress "circular toctree references detected" warnings using `suppress_warnings`
- C, added scope control directives, `c:namespace`, `c:namespace-push`, and `c:namespace-pop`.
- [#2044](#)¹⁹⁹⁵: `autodoc`: Suppress default value for instance attributes
- [#7473](#)¹⁹⁹⁶: `autodoc`: consider a member public if docstring contains `:meta public:` in `info-field-list`

¹⁹⁹³ <https://github.com/sphinx-doc/sphinx/issues/7477>

¹⁹⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/7410>

¹⁹⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/2044>

¹⁹⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/7473>

- #7487¹⁹⁹⁷: autodoc: Allow to generate docs for singledispatch functions by py:autofunction
- #7143¹⁹⁹⁸: autodoc: Support final classes and methods
- #7384¹⁹⁹⁹: autodoc: Support signatures defined by `__new__()`, metaclasses and builtin base classes
- #2106²⁰⁰⁰: autodoc: Support multiple signatures on docstring
- #4422²⁰⁰¹: autodoc: Support GenericAlias in Python 3.7 or above
- #3610²⁰⁰²: autodoc: Support overloaded functions
- #7722²⁰⁰³: autodoc: Support TypeVar
- #7466²⁰⁰⁴: autosummary: headings in generated documents are not translated
- #7490²⁰⁰⁵: autosummary: Add `:caption:` option to autosummary directive to set a caption to the toctree
- #7469²⁰⁰⁶: autosummary: Support module attributes
- #248²⁰⁰⁷, #6040²⁰⁰⁸: autosummary: Add `:recursive:` option to autosummary directive to generate stub files recursively
- #4030²⁰⁰⁹: autosummary: Add `autosummary_context` to add template variables for custom templates
- #7530²⁰¹⁰: html: Support nested `<kbd>` elements
- #7481²⁰¹¹: html theme: Add right margin to footnote/citation labels
- #7482²⁰¹², #7717²⁰¹³: html theme: CSS spacing for code blocks with captions and line numbers
- #7443²⁰¹⁴: html theme: Add new options `globaltoc_collapse` and `globaltoc_includehidden` to control the behavior of globaltoc in sidebar
- #7484²⁰¹⁵: html theme: Avoid clashes between sidebar and other blocks

¹⁹⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/7487>

¹⁹⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/7143>

¹⁹⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/7384>

²⁰⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/2106>

²⁰⁰¹ <https://github.com/sphinx-doc/sphinx/issues/4422>

²⁰⁰² <https://github.com/sphinx-doc/sphinx/issues/3610>

²⁰⁰³ <https://github.com/sphinx-doc/sphinx/issues/7722>

²⁰⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/7466>

²⁰⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/7490>

²⁰⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/7469>

²⁰⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/248>

²⁰⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/6040>

²⁰⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/4030>

²⁰¹⁰ <https://github.com/sphinx-doc/sphinx/issues/7530>

²⁰¹¹ <https://github.com/sphinx-doc/sphinx/issues/7481>

²⁰¹² <https://github.com/sphinx-doc/sphinx/issues/7482>

²⁰¹³ <https://github.com/sphinx-doc/sphinx/issues/7717>

²⁰¹⁴ <https://github.com/sphinx-doc/sphinx/issues/7443>

²⁰¹⁵ <https://github.com/sphinx-doc/sphinx/issues/7484>

- #7476²⁰¹⁶: html theme: Relbar breadcrumb should contain current page
- #7506²⁰¹⁷: html theme: A canonical URL is not escaped
- #7533²⁰¹⁸: html theme: Avoid whitespace at the beginning of genindex.html
- #7541²⁰¹⁹: html theme: Add a "clearer" at the end of the "body"
- #7542²⁰²⁰: html theme: Make admonition/topic/sidebar scrollable
- #7543²⁰²¹: html theme: Add top and bottom margins to tables
- #7695²⁰²²: html theme: Add viewport meta tag for basic theme
- #7721²⁰²³: html theme: classic: default codetextcolor/codebgcolor doesn't override Pygments
- C and C++: allow semicolon in the end of declarations.
- C++, parse parameterized noexcept specifiers.
- #7294²⁰²⁴: C++, parse expressions with user-defined literals.
- C++, parse trailing return types.
- #7143²⁰²⁵: py domain: Add `:final:` option to `py:class`, `py:exception` and `py:method` directives
- #7596²⁰²⁶: py domain: Change a type annotation for variables to a hyperlink
- #7770²⁰²⁷: std domain: `option` directive support arguments in the form of `foo[=bar]`
- #7582²⁰²⁸: napoleon: a type for attribute are represented like type annotation
- #7734²⁰²⁹: napoleon: overescaped trailing underscore on attribute
- #7247²⁰³⁰: linkcheck: Add `linkcheck_request_headers` to send custom HTTP headers for specific host
- #7792²⁰³¹: setuptools: Support `--verbosity` option
- #7683²⁰³²: Add `allowed_exceptions` parameter to `Sphinx.emit()` to allow handlers to raise specified exceptions

²⁰¹⁶ <https://github.com/sphinx-doc/sphinx/issues/7476>

²⁰¹⁷ <https://github.com/sphinx-doc/sphinx/issues/7506>

²⁰¹⁸ <https://github.com/sphinx-doc/sphinx/issues/7533>

²⁰¹⁹ <https://github.com/sphinx-doc/sphinx/issues/7541>

²⁰²⁰ <https://github.com/sphinx-doc/sphinx/issues/7542>

²⁰²¹ <https://github.com/sphinx-doc/sphinx/issues/7543>

²⁰²² <https://github.com/sphinx-doc/sphinx/issues/7695>

²⁰²³ <https://github.com/sphinx-doc/sphinx/issues/7721>

²⁰²⁴ <https://github.com/sphinx-doc/sphinx/issues/7294>

²⁰²⁵ <https://github.com/sphinx-doc/sphinx/issues/7143>

²⁰²⁶ <https://github.com/sphinx-doc/sphinx/issues/7596>

²⁰²⁷ <https://github.com/sphinx-doc/sphinx/issues/7770>

²⁰²⁸ <https://github.com/sphinx-doc/sphinx/issues/7582>

²⁰²⁹ <https://github.com/sphinx-doc/sphinx/issues/7734>

²⁰³⁰ <https://github.com/sphinx-doc/sphinx/issues/7247>

²⁰³¹ <https://github.com/sphinx-doc/sphinx/issues/7792>

²⁰³² <https://github.com/sphinx-doc/sphinx/issues/7683>

- #7295²⁰³³: C++, parse (trailing) requires clauses.

Bugs fixed

- #6703²⁰³⁴: autodoc: incremental build does not work for imported objects
- #7564²⁰³⁵: autodoc: annotations not to be shown for descriptors
- #6588²⁰³⁶: autodoc: Decorated inherited method has no documentation
- #7469²⁰³⁷: autodoc: The change of autodoc-process-docstring for variables is cached unexpectedly
- #7559²⁰³⁸: autodoc: misdetects a sync function is async
- #6857²⁰³⁹: autodoc: failed to detect a classmethod on Enum class
- #7562²⁰⁴⁰: autodoc: a typehint contains spaces is wrongly rendered under `autodoc_typehints='description'` mode
- #7551²⁰⁴¹: autodoc: failed to import nested class
- #7362²⁰⁴²: autodoc: does not render correct signatures for built-in functions
- #7654²⁰⁴³: autodoc: `Optional[Union[foo, bar]]` is presented as `Union[foo, bar, None]`
- #7629²⁰⁴⁴: autodoc: autofunction emits an unfriendly warning if an invalid object specified
- #7650²⁰⁴⁵: autodoc: undecorated signature is shown for decorated functions
- #7676²⁰⁴⁶: autodoc: typo in the default value of `autodoc_member_order`
- #7676²⁰⁴⁷: autodoc: wrong value for `:member-order:` option is ignored silently
- #7676²⁰⁴⁸: autodoc: `member-order="bysource"` does not work for C module
- #3673²⁰⁴⁹: autodoc: `member-order="bysource"` does not work for a module having `__all__`

²⁰³³ <https://github.com/sphinx-doc/sphinx/issues/7295>

²⁰³⁴ <https://github.com/sphinx-doc/sphinx/issues/6703>

²⁰³⁵ <https://github.com/sphinx-doc/sphinx/issues/7564>

²⁰³⁶ <https://github.com/sphinx-doc/sphinx/issues/6588>

²⁰³⁷ <https://github.com/sphinx-doc/sphinx/issues/7469>

²⁰³⁸ <https://github.com/sphinx-doc/sphinx/issues/7559>

²⁰³⁹ <https://github.com/sphinx-doc/sphinx/issues/6857>

²⁰⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/7562>

²⁰⁴¹ <https://github.com/sphinx-doc/sphinx/issues/7551>

²⁰⁴² <https://github.com/sphinx-doc/sphinx/issues/7362>

²⁰⁴³ <https://github.com/sphinx-doc/sphinx/issues/7654>

²⁰⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/7629>

²⁰⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/7650>

²⁰⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/7676>

²⁰⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/7676>

²⁰⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/7676>

²⁰⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/3673>

- [#7668²⁰⁵⁰](#): autodoc: wrong retann value is passed to a handler of autodoc-process-signature
- [#7711²⁰⁵¹](#): autodoc: fails with ValueError when processing numpy objects
- [#7791²⁰⁵²](#): autodoc: TypeError is raised on documenting singledispatch function
- [#7551²⁰⁵³](#): autosummary: a nested class is indexed as non-nested class
- [#7661²⁰⁵⁴](#): autosummary: autosummary directive emits warnings twice if failed to import the target module
- [#7685²⁰⁵⁵](#): autosummary: The template variable "members" contains imported members even if `autosummary_imported_members` is False
- [#7671²⁰⁵⁶](#): autosummary: The location of import failure warning is missing
- [#7535²⁰⁵⁷](#): sphinx-autogen: crashes when custom template uses inheritance
- [#7536²⁰⁵⁸](#): sphinx-autogen: crashes when template uses `il8n` feature
- [#7781²⁰⁵⁹](#): sphinx-build: Wrong error message when outdir is not directory
- [#7653²⁰⁶⁰](#): sphinx-quickstart: Fix multiple directory creation for nested relpath
- [#2785²⁰⁶¹](#): html: Bad alignment of equation links
- [#7718²⁰⁶²](#): html theme: some themes does not respect background color of Pygments style (agogo, haiku, nature, pyramid, scrolls, sphinxdoc and traditional)
- [#7544²⁰⁶³](#): html theme: inconsistent padding in admonitions
- [#7581²⁰⁶⁴](#): napoleon: bad parsing of inline code in attribute docstrings
- [#7628²⁰⁶⁵](#): imgconverter: runs imagemagick once unnecessary for builders not supporting images
- [#7610²⁰⁶⁶](#): incorrectly renders consecutive backslashes for Docutils 0.16
- [#7646²⁰⁶⁷](#): handle errors on event handlers

²⁰⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/7668>

²⁰⁵¹ <https://github.com/sphinx-doc/sphinx/issues/7711>

²⁰⁵² <https://github.com/sphinx-doc/sphinx/issues/7791>

²⁰⁵³ <https://github.com/sphinx-doc/sphinx/issues/7551>

²⁰⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/7661>

²⁰⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/7685>

²⁰⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/7671>

²⁰⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/7535>

²⁰⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/7536>

²⁰⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/7781>

²⁰⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/7653>

²⁰⁶¹ <https://github.com/sphinx-doc/sphinx/issues/2785>

²⁰⁶² <https://github.com/sphinx-doc/sphinx/issues/7718>

²⁰⁶³ <https://github.com/sphinx-doc/sphinx/issues/7544>

²⁰⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/7581>

²⁰⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/7628>

²⁰⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/7610>

²⁰⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/7646>

- #4187²⁰⁶⁸: LaTeX: EN DASH disappears from PDF bookmarks in Japanese documents
- #7701²⁰⁶⁹: LaTeX: Anonymous indirect hyperlink target causes duplicated labels
- #7723²⁰⁷⁰: LaTeX: pdflatex crashed when URL contains a single quote
- #7756²⁰⁷¹: py domain: The default value for positional only argument is not shown
- #7760²⁰⁷²: coverage: Add `coverage_show_missing_items` to show coverage result to console
- C++, fix rendering and xrefs in nested names explicitly starting in global scope, e.g., `::A::B`.
- C, fix rendering and xrefs in nested names explicitly starting in global scope, e.g., `.A.B`.
- #7763²⁰⁷³: C and C++, don't crash during display stringification of unary expressions and fold expressions.

14.71 Release 3.0.4 (released May 27, 2020)

Bugs fixed

- #7567²⁰⁷⁴: autodoc: parametrized types are shown twice for generic types
- #7637²⁰⁷⁵: autodoc: system defined TypeVars are shown in Python 3.9
- #7696²⁰⁷⁶: html: Updated jQuery version from 3.4.1 to 3.5.1 for security reasons
- #7611²⁰⁷⁷: md5 fails when OpenSSL FIPS is enabled
- #7626²⁰⁷⁸: release package does not contain CODE_OF_CONDUCT

²⁰⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/4187>

²⁰⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/7701>

²⁰⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/7723>

²⁰⁷¹ <https://github.com/sphinx-doc/sphinx/issues/7756>

²⁰⁷² <https://github.com/sphinx-doc/sphinx/issues/7760>

²⁰⁷³ <https://github.com/sphinx-doc/sphinx/issues/7763>

²⁰⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/7567>

²⁰⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/7637>

²⁰⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/7696>

²⁰⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/7611>

²⁰⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/7626>

14.72 Release 3.0.3 (released Apr 26, 2020)

Funzionalità aggiunte

- C, parse array declarators with static, qualifiers, and VLA specification.

Bugs fixed

- [#7516](#)²⁰⁷⁹: autodoc: crashes if target object raises an error on accessing its attributes

14.73 Release 3.0.2 (released Apr 19, 2020)

Funzionalità aggiunte

- C, parse attributes and add `c_id_attributes` and `c_paren_attributes` to support user-defined attributes.

Bugs fixed

- [#7461](#)²⁰⁸⁰: py domain: fails with IndexError for empty tuple in type annotation
- [#7510](#)²⁰⁸¹: py domain: keyword-only arguments are documented as having a default of None
- [#7418](#)²⁰⁸²: std domain: `term` role could not match case-insensitively
- [#7461](#)²⁰⁸³: autodoc: empty tuple in type annotation is not shown correctly
- [#7479](#)²⁰⁸⁴: autodoc: Sphinx builds has been slower since 3.0.0 on mocking
- C++, fix spacing issue in east-const declarations.
- [#7414](#)²⁰⁸⁵: LaTeX: Xindy language options were incorrect
- Sphinx crashes with ImportError on python3.5.1

²⁰⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/7516>

²⁰⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/7461>

²⁰⁸¹ <https://github.com/sphinx-doc/sphinx/issues/7510>

²⁰⁸² <https://github.com/sphinx-doc/sphinx/issues/7418>

²⁰⁸³ <https://github.com/sphinx-doc/sphinx/issues/7461>

²⁰⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/7479>

²⁰⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/7414>

14.74 Release 3.0.1 (released Apr 11, 2020)

Modifiche incompatibili

- #7418²⁰⁸⁶: std domain: *term* role becomes case sensitive

Bugs fixed

- #7428²⁰⁸⁷: py domain: a reference to class None emits a nitpicky warning
- #7445²⁰⁸⁸: py domain: a return annotation None in the function signature is not converted to a hyperlink when using intersphinx
- #7418²⁰⁸⁹: std domain: duplication warning for glossary terms is case insensitive
- #7438²⁰⁹⁰: C++, fix merging overloaded functions in parallel builds.
- #7422²⁰⁹¹: autodoc: fails with ValueError when using autodoc_mock_imports
- #7435²⁰⁹²: autodoc: *autodoc_typehints*='description' doesn't suppress typehints in signature for classes/methods
- #7451²⁰⁹³: autodoc: fails with AttributeError when an object returns non-string object as a *__doc__* member
- #7423²⁰⁹⁴: crashed when giving a non-string object to logger
- #7479²⁰⁹⁵: html theme: Do not include xmlns attribute with HTML 5 doctype
- #7426²⁰⁹⁶: html theme: Escape some links in HTML templates

14.75 Release 3.0.0 (released Apr 06, 2020)

Dipendenze

3.0.0b1

²⁰⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/7418>

²⁰⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/7428>

²⁰⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/7445>

²⁰⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/7418>

²⁰⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/7438>

²⁰⁹¹ <https://github.com/sphinx-doc/sphinx/issues/7422>

²⁰⁹² <https://github.com/sphinx-doc/sphinx/issues/7435>

²⁰⁹³ <https://github.com/sphinx-doc/sphinx/issues/7451>

²⁰⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/7423>

²⁰⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/7479>

²⁰⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/7426>

- LaTeX: drop dependency on **extractbb** for image inclusion in Japanese documents as .xbb files are unneeded by **dvipdfmx** since TeXLive2015 (refs: [#6189](#)²⁰⁹⁷)
- babel-2.0 or above is available (Unpinned)

Modifiche incompatibili

3.0.0b1

- Drop features and APIs deprecated in 1.8.x
- [#247](#)²⁰⁹⁸: autosummary: stub files are overwritten automatically by default. see [autosummary_generate_overwrite](#) to change the behavior
- [#5923](#)²⁰⁹⁹: autodoc: the members of object class are not documented by default when :inherited-members: and :special-members: are given.
- [#6830](#)²¹⁰⁰: py domain: meta fields in info-field-list becomes reserved. They are not displayed on output document now
- [#6417](#)²¹⁰¹: py domain: doctree of desc_parameterlist has been changed. The argument names, annotations and default values are wrapped with inline node
- The structure of `sphinx.events.EventManager.listeners` has changed
- Due to the scoping changes for [productionlist](#) some uses of [token](#) must be modified to include the scope which was previously ignored.
- [#6903](#)²¹⁰²: Internal data structure of Python, reST and standard domains have changed. The node_id is added to the index of objects and modules. Now they contains a pair of docname and node_id for cross reference.
- [#7276](#)²¹⁰³: C++ domain: Non intended behavior is removed such as `say_hello_` links to `.. cpp:function:: say_hello()`
- [#7210](#)²¹⁰⁴: js domain: Non intended behavior is removed such as `parseInt_` links to `.. js:function:: parseInt`
- [#7229](#)²¹⁰⁵: rst domain: Non intended behavior is removed such as `numref_` links to `.. rst:role:: numref`
- [#6903](#)²¹⁰⁶: py domain: Non intended behavior is removed such as `say_hello_` links to `.. py:function:: say_hello()`

²⁰⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/6189>

²⁰⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/247>

²⁰⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/5923>

²¹⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/6830>

²¹⁰¹ <https://github.com/sphinx-doc/sphinx/issues/6417>

²¹⁰² <https://github.com/sphinx-doc/sphinx/issues/6903>

²¹⁰³ <https://github.com/sphinx-doc/sphinx/issues/7276>

²¹⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/7210>

²¹⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/7229>

²¹⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/6903>

- [#7246](#)²¹⁰⁷: py domain: Drop special cross reference helper for exceptions, functions and methods
- The C domain has been rewritten, with additional directives and roles. The existing ones are now more strict, resulting in new warnings.
- The attribute `sphinx_cpp_tagname` in the `desc_signature_line` node has been renamed to `sphinx_line_type`.
- [#6462](#)²¹⁰⁸: double backslashes in domain directives are no longer replaced by single backslashes as default. A new configuration value `strip_signature_backslash` can be used by users to re-enable it.

3.0.0 final

- [#7222](#)²¹⁰⁹: `sphinx.util.inspect.unwrap()` is renamed to `unwrap_all()`

Deprecato

3.0.0b1

- `desc_signature['first']`
- `sphinx.directives.DescDirective`
- `sphinx.domains.std.StandardDomain.add_object()`
- `sphinx.domains.python.PyDecoratorMixin`
- `sphinx.ext.autodoc.get_documenters()`
- `sphinx.ext.autosummary.process_autosummary_toc()`
- `sphinx.parsers.Parser.app`
- `sphinx.testing.path.Path.text()`
- `sphinx.testing.path.Path.bytes()`
- `sphinx.util.inspect.getargspec()`
- `sphinx.writers.latex.LaTeXWriter.format_docclass()`

²¹⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/7246>

²¹⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/6462>

²¹⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/7222>

Funzionalità aggiunte

3.0.0b1

- #247²¹¹⁰: autosummary: Add `autosummary_generate_overwrite` to overwrite old stub file
- #5923²¹¹¹: autodoc: `:inherited-members:` option takes a name of ancestor class not to document inherited members of the class and uppers
- #6830²¹¹²: autodoc: consider a member private if docstring contains `:meta private:` in info-field-list
- #7165²¹¹³: autodoc: Support Annotated type (PEP-593)
- #2815²¹¹⁴: autodoc: Support singledispatch functions and methods
- #7079²¹¹⁵: autodoc: `autodoc_typehints` accepts "description" configuration. It shows typehints as object description
- #7314²¹¹⁶: apidoc: Propagate `--maxdepth` option through package documents
- #6558²¹¹⁷: glossary: emit a warning for duplicated glossary entry
- #3106²¹¹⁸: domain: Register hyperlink target for index page automatically
- #6558²¹¹⁹: std domain: emit a warning for duplicated generic objects
- #6830²¹²⁰: py domain: Add new event: `object-description-transform`
- #6895²¹²¹: py domain: Do not emit nitpicky warnings for built-in types
- py domain: Support lambda functions in function signature
- #6417²¹²²: py domain: Allow to make a style for arguments of functions and methods
- #7238²¹²³, #7239²¹²⁴: py domain: Emit a warning on describing a python object if the entry is already added as the same name
- #7341²¹²⁵: py domain: type annotations in signature are converted to cross refs
- Support priority of event handlers. For more detail, see `Sphinx.connect()`

²¹¹⁰ <https://github.com/sphinx-doc/sphinx/issues/247>

²¹¹¹ <https://github.com/sphinx-doc/sphinx/issues/5923>

²¹¹² <https://github.com/sphinx-doc/sphinx/issues/6830>

²¹¹³ <https://github.com/sphinx-doc/sphinx/issues/7165>

²¹¹⁴ <https://github.com/sphinx-doc/sphinx/issues/2815>

²¹¹⁵ <https://github.com/sphinx-doc/sphinx/issues/7079>

²¹¹⁶ <https://github.com/sphinx-doc/sphinx/issues/7314>

²¹¹⁷ <https://github.com/sphinx-doc/sphinx/issues/6558>

²¹¹⁸ <https://github.com/sphinx-doc/sphinx/issues/3106>

²¹¹⁹ <https://github.com/sphinx-doc/sphinx/issues/6558>

²¹²⁰ <https://github.com/sphinx-doc/sphinx/issues/6830>

²¹²¹ <https://github.com/sphinx-doc/sphinx/issues/6895>

²¹²² <https://github.com/sphinx-doc/sphinx/issues/6417>

²¹²³ <https://github.com/sphinx-doc/sphinx/issues/7238>

²¹²⁴ <https://github.com/sphinx-doc/sphinx/issues/7239>

²¹²⁵ <https://github.com/sphinx-doc/sphinx/issues/7341>

- #3077²¹²⁶: Implement the scoping for *productionlist* as indicated in the documentation.
- #1027²¹²⁷: Support backslash line continuation in *productionlist*.
- #7108²¹²⁸: config: Allow to show an error message from conf.py via ConfigError
- #7032²¹²⁹: html: *html_scaled_image_link* will be disabled for images having *no-scaled-link* class
- #7144²¹³⁰: Add CSS class indicating its domain for each desc node
- #7211²¹³¹: latex: Use babel for Chinese document when using XeLaTeX
- #6672²¹³²: LaTeX: Support LaTeX Theming (experimental)
- #7005²¹³³: LaTeX: Add LaTeX styling macro for *kbd* role
- #7220²¹³⁴: genindex: Show "main" index entries at first
- #7103²¹³⁵: linkcheck: writes all links to output.json
- #7025²¹³⁶: html search: full text search can be disabled for individual document using *:nosearch:* file-wide metadata
- #7293²¹³⁷: html search: Allow to override JavaScript splitter via *SearchLanguage.js_splitter_code*
- #7142²¹³⁸: html theme: Add a theme option: *pygments_dark_style* to switch the style of code-blocks in dark mode
- The C domain has been rewritten adding for example:
 - Cross-referencing respecting the current scope.
 - Possible to document anonymous entities.
 - More specific directives and roles for each type of entity, e.g., handling scoping of enumerators.
 - New role *c:expr* for rendering expressions and types in text.
- Added `SphinxDirective.get_source_info()` and `SphinxRole.get_source_info()`.

²¹²⁶ <https://github.com/sphinx-doc/sphinx/issues/3077>

²¹²⁷ <https://github.com/sphinx-doc/sphinx/issues/1027>

²¹²⁸ <https://github.com/sphinx-doc/sphinx/issues/7108>

²¹²⁹ <https://github.com/sphinx-doc/sphinx/issues/7032>

²¹³⁰ <https://github.com/sphinx-doc/sphinx/issues/7144>

²¹³¹ <https://github.com/sphinx-doc/sphinx/issues/7211>

²¹³² <https://github.com/sphinx-doc/sphinx/issues/6672>

²¹³³ <https://github.com/sphinx-doc/sphinx/issues/7005>

²¹³⁴ <https://github.com/sphinx-doc/sphinx/issues/7220>

²¹³⁵ <https://github.com/sphinx-doc/sphinx/issues/7103>

²¹³⁶ <https://github.com/sphinx-doc/sphinx/issues/7025>

²¹³⁷ <https://github.com/sphinx-doc/sphinx/issues/7293>

²¹³⁸ <https://github.com/sphinx-doc/sphinx/issues/7142>

- [#7324](#)²¹³⁹: sphinx-build: Emit a warning if multiple files having different file extensions for same document found

3.0.0 final

- Added `ObjectDescription.transform_content()`.

Bugs fixed

3.0.0b1

- C++, fix cross reference lookup in certain cases involving function overloads.
- [#5078](#)²¹⁴⁰: C++, fix cross reference lookup when a directive contains multiple declarations.
- C++, suppress warnings for directly dependent typenames in cross references generated automatically in signatures.
- [#5637](#)²¹⁴¹: autodoc: Incorrect handling of nested class names on show-inheritance
- [#7267](#)²¹⁴²: autodoc: error message for invalid directive options has wrong location
- [#7329](#)²¹⁴³: autodoc: info-field-list is wrongly generated from type hints into the class description even if `autoclass_content='class'` set
- [#7331](#)²¹⁴⁴: autodoc: a cython-function is not recognized as a function
- [#5637](#)²¹⁴⁵: inheritance_diagram: Incorrect handling of nested class names
- [#7139](#)²¹⁴⁶: code-block:: guess does not work
- [#7325](#)²¹⁴⁷: html: source_suffix containing dot leads to wrong source link
- [#7357](#)²¹⁴⁸: html: Resizing SVG image fails with ValueError
- [#7278](#)²¹⁴⁹: html search: Fix use of `html_file_suffix` instead of `html_link_suffix` in search results
- [#7297](#)²¹⁵⁰: html theme: bizstyle does not support sidebarwidth
- [#3842](#)²¹⁵¹: singlehtml: Path to images broken when master doc is not in source root

²¹³⁹ <https://github.com/sphinx-doc/sphinx/issues/7324>

²¹⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/5078>

²¹⁴¹ <https://github.com/sphinx-doc/sphinx/issues/5637>

²¹⁴² <https://github.com/sphinx-doc/sphinx/issues/7267>

²¹⁴³ <https://github.com/sphinx-doc/sphinx/issues/7329>

²¹⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/7331>

²¹⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/5637>

²¹⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/7139>

²¹⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/7325>

²¹⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/7357>

²¹⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/7278>

²¹⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/7297>

²¹⁵¹ <https://github.com/sphinx-doc/sphinx/issues/3842>

- #7179²¹⁵²: std domain: Fix whitespaces are suppressed on referring GenericObject
- #7289²¹⁵³: console: use bright colors instead of bold
- #1539²¹⁵⁴: C, parse array types.
- #2377²¹⁵⁵: C, parse function pointers even in complex types.
- #7345²¹⁵⁶: sphinx-build: Sphinx crashes if output directory exists as a file
- #7290²¹⁵⁷: sphinx-build: Ignore bdb.BdbQuit when handling exceptions
- #6240²¹⁵⁸: napoleon: Attributes and Methods sections ignore :noindex: option

3.0.0 final

- #7364²¹⁵⁹: autosummary: crashed when `autosummary_generate` is False
- #7370²¹⁶⁰: autosummary: raises UnboundLocalError when unknown module given
- #7367²¹⁶¹: C++, alternate operator spellings are now supported.
- C, alternate operator spellings are now supported.
- #7368²¹⁶²: C++, comma operator in expressions, pack expansion in template argument lists, and more comprehensive error messages in some cases.
- C, C++, fix crash and wrong duplicate warnings related to anon symbols.
- #6477²¹⁶³: Escape first "!" in a cross reference linking no longer possible
- #7219²¹⁶⁴: py domain: The index entry generated by `py:function` directive is different with one from `index` directive with "builtin" type
- #7301²¹⁶⁵: capital characters are not allowed for `node_id`
- #7301²¹⁶⁶: epub: duplicated `node_ids` are generated
- #6564²¹⁶⁷: html: a width of table was ignored on HTML builder
- #7401²¹⁶⁸: Incorrect argument is passed for `env-get-outdated` handlers

²¹⁵² <https://github.com/sphinx-doc/sphinx/issues/7179>

²¹⁵³ <https://github.com/sphinx-doc/sphinx/issues/7289>

²¹⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/1539>

²¹⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/2377>

²¹⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/7345>

²¹⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/7290>

²¹⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/6240>

²¹⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/7364>

²¹⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/7370>

²¹⁶¹ <https://github.com/sphinx-doc/sphinx/issues/7367>

²¹⁶² <https://github.com/sphinx-doc/sphinx/issues/7368>

²¹⁶³ <https://github.com/sphinx-doc/sphinx/issues/6477>

²¹⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/7219>

²¹⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/7301>

²¹⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/7301>

²¹⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/6564>

²¹⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/7401>

- [#7355](#)²¹⁶⁹: autodoc: a signature of cython-function is not recognized well
- [#7222](#)²¹⁷⁰: autodoc: `__wrapped__` functions are not documented correctly
- [#7409](#)²¹⁷¹: intersphinx: `ValueError` is raised when an extension sets up *intersphinx_mapping* on *config-inited* event
- [#7343](#)²¹⁷²: Sphinx builds has been slower since 2.4.0 on debug mode

14.76 Release 2.4.5 (released Nov 18, 2021)

Dipendenze

- [#9807](#)²¹⁷³: Restrict Docutils to 0.17.x or older

14.77 Release 2.4.4 (released Mar 05, 2020)

Bugs fixed

- [#7197](#)²¹⁷⁴: LaTeX: `platex` cause error to build image directive with target url
- [#7223](#)²¹⁷⁵: Sphinx builds has been slower since 2.4.0

14.78 Release 2.4.3 (released Feb 22, 2020)

Bugs fixed

- [#7184](#)²¹⁷⁶: autodoc: `*args` and `**kwarg` in type comments are not handled properly
- [#7189](#)²¹⁷⁷: autodoc: classmethod coroutines are not detected
- [#7183](#)²¹⁷⁸: intersphinx: `:attr:` reference to property is broken
- [#6244](#)²¹⁷⁹, [#6387](#)²¹⁸⁰: html search: Search breaks/hangs when built with `dirhtml` builder

²¹⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/7355>

²¹⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/7222>

²¹⁷¹ <https://github.com/sphinx-doc/sphinx/issues/7409>

²¹⁷² <https://github.com/sphinx-doc/sphinx/issues/7343>

²¹⁷³ <https://github.com/sphinx-doc/sphinx/issues/9807>

²¹⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/7197>

²¹⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/7223>

²¹⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/7184>

²¹⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/7189>

²¹⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/7183>

²¹⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/6244>

²¹⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/6387>

- #7195²¹⁸¹: todo: emit doctree-resolved event with non-document node incorrectly

14.79 Release 2.4.2 (released Feb 19, 2020)

Bugs fixed

- #7138²¹⁸²: autodoc: `autodoc.typehints` crashed when variable has unbound object as a value
- #7156²¹⁸³: autodoc: separator for keyword only arguments is not shown
- #7146²¹⁸⁴: autodoc: `IndexError` is raised on suppressed `type_comment` found
- #7161²¹⁸⁵: autodoc: `typehints` extension does not support parallel build
- #7178²¹⁸⁶: autodoc: `TypeError` is raised on fetching type annotations
- #7151²¹⁸⁷: crashed when extension assigns a value to `env.indexentries`
- #7170²¹⁸⁸: text: Remove debug print
- #7137²¹⁸⁹: viewcode: Avoid to crash when non-python code given

14.80 Release 2.4.1 (released Feb 11, 2020)

Bugs fixed

- #7120²¹⁹⁰: html: crashed when on scaling SVG images which have float dimensions
- #7126²¹⁹¹: autodoc: `TypeError: 'getset_descriptor' object is not iterable`

²¹⁸¹ <https://github.com/sphinx-doc/sphinx/issues/7195>

²¹⁸² <https://github.com/sphinx-doc/sphinx/issues/7138>

²¹⁸³ <https://github.com/sphinx-doc/sphinx/issues/7156>

²¹⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/7146>

²¹⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/7161>

²¹⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/7178>

²¹⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/7151>

²¹⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/7170>

²¹⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/7137>

²¹⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/7120>

²¹⁹¹ <https://github.com/sphinx-doc/sphinx/issues/7126>

14.81 Release 2.4.0 (released Feb 09, 2020)

Deprecato

- The `decode` argument of `sphinx.pycode.ModuleAnalyzer()`
- `sphinx.directives.other.Index`
- `sphinx.environment.temp_data['gloss_entries']`
- `sphinx.environment.BuildEnvironment.indexentries`
- `sphinx.environment.collectors.indexentries.IndexEntriesCollector`
- `sphinx.ext.apidoc.INITPY`
- `sphinx.ext.apidoc.shall_skip()`
- `sphinx.io.FiletypeNotFoundError`
- `sphinx.io.get_filetype()`
- `sphinx.pycode.ModuleAnalyzer.encoding`
- `sphinx.roles.Index`
- `sphinx.util.detect_encoding()`
- `sphinx.util.get_module_source()`
- `sphinx.util.inspect.Signature`
- `sphinx.util.inspect.safe_getmembers()`
- `sphinx.writers.latex.LaTeXTranslator.settings.author`
- `sphinx.writers.latex.LaTeXTranslator.settings.contentsname`
- `sphinx.writers.latex.LaTeXTranslator.settings.docclass`
- `sphinx.writers.latex.LaTeXTranslator.settings.docname`
- `sphinx.writers.latex.LaTeXTranslator.settings.title`
- `sphinx.writers.latex.ADDITIONAL_SETTINGS`
- `sphinx.writers.latex.DEFAULT_SETTINGS`
- `sphinx.writers.latex.LUALATEX_DEFAULT_FONTPKG`
- `sphinx.writers.latex.PDFLATEX_DEFAULT_FONTPKG`
- `sphinx.writers.latex.XELATEX_DEFAULT_FONTPKG`
- `sphinx.writers.latex.XELATEX_GREEK_DEFAULT_FONTPKG`

Funzionalità aggiunte

- [#6910²¹⁹²](#): `inheritance_diagram`: Make the background of diagrams transparent
- [#6446²¹⁹³](#): `duration`: Add `sphinx.ext.durations` to inspect which documents slow down the build
- [#6837²¹⁹⁴](#): LaTeX: Support a nested table
- [#7115²¹⁹⁵](#): LaTeX: Allow to override `LATEXOPTS` and `LATEXMKOPTS` via environment variable
- [#6966²¹⁹⁶](#): `graphviz`: Support `:class:` option
- [#6696²¹⁹⁷](#): `html`: `:scale:` option of `image/figure` directive not working for SVG images (`imagesize-1.2.0` or above is required)
- [#6994²¹⁹⁸](#): `imgconverter`: Support illustrator file (.ai) to .png conversion
- `autodoc`: Support Positional-Only Argument separator (PEP-570 compliant)
- `autodoc`: Support type annotations for variables
- [#2755²¹⁹⁹](#): `autodoc`: Add new event: *`autodoc-before-process-signature`*
- [#2755²²⁰⁰](#): `autodoc`: Support `type_comment` style (ex. `# type: (str) -> str`) annotation (python3.8+ or `typed_ast2201` is required)
- [#7051²²⁰²](#): `autodoc`: Support instance variables without defaults (PEP-526)
- [#6418²²⁰³](#): `autodoc`: Add a new extension `sphinx.ext.autodoc.typehints`. It shows typehints as object description if `autodoc_typehints = "description"` set. This is an experimental extension and it will be integrated into `autodoc` core in Sphinx 3.0
- `SphinxTranslator` now calls `visitor/departure` method for super node class if `visitor/departure` method for original node class not found
- [#6418²²⁰⁴](#): Add new event: *`object-description-transform`*
- `py` domain: *`py:data`* and *`py:attribute`* take new options named `:type:` and `:value:` to describe its type and initial value
- [#6785²²⁰⁵](#): `py` domain: `:py:attr:` is able to refer properties again

²¹⁹² <https://github.com/sphinx-doc/sphinx/issues/6910>

²¹⁹³ <https://github.com/sphinx-doc/sphinx/issues/6446>

²¹⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/6837>

²¹⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/7115>

²¹⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/6966>

²¹⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/6696>

²¹⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/6994>

²¹⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/2755>

²²⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/2755>

²²⁰¹ https://github.com/python/typed_ast

²²⁰² <https://github.com/sphinx-doc/sphinx/issues/7051>

²²⁰³ <https://github.com/sphinx-doc/sphinx/issues/6418>

²²⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/6418>

²²⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/6785>

- [#6772²²⁰⁶](#): apidoc: Add `-q` option for quiet mode

Bugs fixed

- [#6925²²⁰⁷](#): html: Remove redundant `type="text/javascript"` from `<script>` elements
- [#7112²²⁰⁸](#): html: SVG image is not layouted as float even if aligned
- [#6906²²⁰⁹](#), [#6907²²¹⁰](#): autodoc: failed to read the source codes encoded in cp1251
- [#6961²²¹¹](#): latex: warning for babel shown twice
- [#7059²²¹²](#): latex: LaTeX compilation falls into infinite loop (wrapfig issue)
- [#6581²²¹³](#): latex: `:reversed:` option for toctree does not effect to LaTeX build
- [#6559²²¹⁴](#): Wrong node-ids are generated in glossary directive
- [#6986²²¹⁵](#): apidoc: misdetects module name for `.so` file inside module
- [#6899²²¹⁶](#): apidoc: private members are not shown even if `--private` given
- [#6327²²¹⁷](#): apidoc: Support a python package consisted of `__init__.so` file
- [#6999²²¹⁸](#): napoleon: fails to parse tilde in `:exc:` role
- [#7019²²¹⁹](#): gettext: Absolute path used in message catalogs
- [#7023²²²⁰](#): autodoc: nested partial functions are not listed
- [#7023²²²¹](#): autodoc: partial functions imported from other modules are listed as module members without `:imported-members:` option
- [#6889²²²²](#): autodoc: Trailing comma in `:members::` option causes cryptic warning
- [#6568²²²³](#): autosummary: `autosummary_imported_members` is ignored on generating a stub file for submodule
- [#7055²²²⁴](#): linkcheck: redirect is treated as an error

²²⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/6772>

²²⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/6925>

²²⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/7112>

²²⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/6906>

²²¹⁰ <https://github.com/sphinx-doc/sphinx/issues/6907>

²²¹¹ <https://github.com/sphinx-doc/sphinx/issues/6961>

²²¹² <https://github.com/sphinx-doc/sphinx/issues/7059>

²²¹³ <https://github.com/sphinx-doc/sphinx/issues/6581>

²²¹⁴ <https://github.com/sphinx-doc/sphinx/issues/6559>

²²¹⁵ <https://github.com/sphinx-doc/sphinx/issues/6986>

²²¹⁶ <https://github.com/sphinx-doc/sphinx/issues/6899>

²²¹⁷ <https://github.com/sphinx-doc/sphinx/issues/6327>

²²¹⁸ <https://github.com/sphinx-doc/sphinx/issues/6999>

²²¹⁹ <https://github.com/sphinx-doc/sphinx/issues/7019>

²²²⁰ <https://github.com/sphinx-doc/sphinx/issues/7023>

²²²¹ <https://github.com/sphinx-doc/sphinx/issues/7023>

²²²² <https://github.com/sphinx-doc/sphinx/issues/6889>

²²²³ <https://github.com/sphinx-doc/sphinx/issues/6568>

²²²⁴ <https://github.com/sphinx-doc/sphinx/issues/7055>

- [#7088](#)²²²⁵: HTML template: If `navigation_with_keys` option is activated, modifier keys are ignored, which means the feature can interfere with browser features
- [#7090](#)²²²⁶: std domain: Can't assign numfig-numbers for custom container nodes
- [#7106](#)²²²⁷: std domain: enumerated nodes are marked as duplicated when extensions call `note_explicit_target()`
- [#7095](#)²²²⁸: dirhtml: Cross references are broken via intersphinx and `:doc:` role
- C++:
 - Don't crash when using the `struct` role in some cases.
 - Don't warn when using the `var/member` role for function parameters.
 - Render call and braced-init expressions correctly.
- [#7097](#)²²²⁹: Filenames of images generated by `sphinx.transforms.post_transforms.images.ImageConverter` or its subclasses (used for latex build) are now sanitized, to prevent broken paths

14.82 Release 2.3.1 (released Dec 22, 2019)

Bugs fixed

- [#6936](#)²²³⁰: sphinx-autogen: raises `AttributeError`

14.83 Release 2.3.0 (released Dec 15, 2019)

Modifiche incompatibili

- [#6742](#)²²³¹: `end-before` option of `literalinclude` directive does not match the first line of the code block.
- [#1331](#)²²³²: Change default User-Agent header to "Sphinx/X.Y.Z requests/X.Y.Z python/X.Y.Z". It can be changed via `user_agent`.
- [#6867](#)²²³³: text: content of admonitions starts after a blank line

²²²⁵ <https://github.com/sphinx-doc/sphinx/issues/7088>

²²²⁶ <https://github.com/sphinx-doc/sphinx/issues/7090>

²²²⁷ <https://github.com/sphinx-doc/sphinx/issues/7106>

²²²⁸ <https://github.com/sphinx-doc/sphinx/issues/7095>

²²²⁹ <https://github.com/sphinx-doc/sphinx/issues/7097>

²²³⁰ <https://github.com/sphinx-doc/sphinx/issues/6936>

²²³¹ <https://github.com/sphinx-doc/sphinx/issues/6742>

²²³² <https://github.com/sphinx-doc/sphinx/issues/1331>

²²³³ <https://github.com/sphinx-doc/sphinx/issues/6867>

Deprecato

- `sphinx.builders.gettext.POHEADER`
- `sphinx.io.SphinxStandaloneReader.app`
- `sphinx.io.SphinxStandaloneReader.env`
- `sphinx.util.texescape.tex_escape_map`
- `sphinx.util.texescape.tex_hl_escape_map_new`
- `sphinx.writers.latex.LaTeXTranslator.no_contractions`

Funzionalità aggiunte

- #6707²²³⁴: C++, support bit-fields.
- #267²²³⁵: html: Eliminate prompt characters of doctest block from copyable text
- #6548²²³⁶: html: Use favicon for OpenSearch if available
- #6729²²³⁷: html theme: agogo theme now supports `rightsidebar` option
- #6780²²³⁸: Add PEP-561 Support
- #6762²²³⁹: latex: Allow to load additional LaTeX packages via `extrapackages` key of `latex_elements`
- #1331²²⁴⁰: Add new config variable: `user_agent`
- #6000²²⁴¹: LaTeX: have backslash also be an inline literal word wrap break character
- #4186²²⁴²: LaTeX: Support upLaTeX as a new `latex_engine` (experimental)
- #6812²²⁴³: Improve a warning message when extensions are not parallel safe
- #6818²²⁴⁴: Improve Intersphinx performance for multiple remote inventories.
- #2546²²⁴⁵: apidoc: .so file support
- #6798²²⁴⁶: autosummary: emit `autodoc-skip-member` event on generating stub file
- #6483²²⁴⁷: i18n: make explicit titles in toctree translatable

²²³⁴ <https://github.com/sphinx-doc/sphinx/issues/6707>

²²³⁵ <https://github.com/sphinx-doc/sphinx/issues/267>

²²³⁶ <https://github.com/sphinx-doc/sphinx/issues/6548>

²²³⁷ <https://github.com/sphinx-doc/sphinx/issues/6729>

²²³⁸ <https://github.com/sphinx-doc/sphinx/issues/6780>

²²³⁹ <https://github.com/sphinx-doc/sphinx/issues/6762>

²²⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/1331>

²²⁴¹ <https://github.com/sphinx-doc/sphinx/issues/6000>

²²⁴² <https://github.com/sphinx-doc/sphinx/issues/4186>

²²⁴³ <https://github.com/sphinx-doc/sphinx/issues/6812>

²²⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/6818>

²²⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/2546>

²²⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/6798>

²²⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/6483>

- #6816²²⁴⁸: linkcheck: Add `linkcheck_auth` option to provide authentication information when doing linkcheck builds
- #6872²²⁴⁹: linkcheck: Handles HTTP 308 Permanent Redirect
- #6613²²⁵⁰: html: Wrap section number in span tag
- #6781²²⁵¹: gettext: Add `gettext_last_translator` and `gettext_language_team` to customize headers of POT file

Bugs fixed

- #6668²²⁵²: LaTeX: Longtable before header has incorrect distance (refs: [latex3/latex2e`#173 <https://github.com/sphinx-doc/sphinx/issues/173>`](https://github.com/sphinx-doc/sphinx/issues/173)²²⁵³)
- #6618²²⁵⁴: LaTeX: Avoid section names at the end of a page
- #6738²²⁵⁵: LaTeX: Do not replace unicode characters by LaTeX macros on unicode supported LaTeX engines: ℓ, \$, €, ∞, ±, →, →, −, superscript and subscript digits go through "as is" (as default OpenType font supports them)
- #6704²²⁵⁶: linkcheck: Be defensive and handle newly defined HTTP error code
- #6806²²⁵⁷: linkcheck: Failure on parsing content
- #6655²²⁵⁸: image URLs containing data: causes gettext builder crashed
- #6584²²⁵⁹: i18n: Error when compiling message catalogs on Hindi
- #6718²²⁶⁰: i18n: KeyError is raised if section title and table title are same
- #6743²²⁶¹: i18n: `rst_prolog` breaks the translation
- #6708²²⁶²: mathbase: Some deprecated functions have removed
- #6709²²⁶³: autodoc: mock object does not work as a class decorator
- #5070²²⁶⁴: epub: Wrong internal href fragment links

²²⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/6816>

²²⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/6872>

²²⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/6613>

²²⁵¹ <https://github.com/sphinx-doc/sphinx/issues/6781>

²²⁵² <https://github.com/sphinx-doc/sphinx/issues/6668>

²²⁵³ <https://github.com/latex3/latex2e/issues/173>

²²⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/6618>

²²⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/6738>

²²⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/6704>

²²⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/6806>

²²⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/6655>

²²⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/6584>

²²⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/6718>

²²⁶¹ <https://github.com/sphinx-doc/sphinx/issues/6743>

²²⁶² <https://github.com/sphinx-doc/sphinx/issues/6708>

²²⁶³ <https://github.com/sphinx-doc/sphinx/issues/6709>

²²⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/5070>

- [#6712²²⁶⁵](#): Allow not to install sphinx.testing as runtime (mainly for ALT Linux)
- [#6741²²⁶⁶](#): html: search result was broken with empty `html_file_suffix`
- [#6001²²⁶⁷](#): LaTeX does not wrap long code lines at backslash character
- [#6804²²⁶⁸](#): LaTeX: PDF build breaks if admonition of danger type contains code-block long enough not to fit on one page
- [#6809²²⁶⁹](#): LaTeX: code-block in a danger type admonition can easily spill over bottom of page
- [#6793²²⁷⁰](#): texinfo: Code examples broken following "sidebar"
- [#6813²²⁷¹](#): An orphan warning is emitted for included document on Windows. Thanks to @drillan
- [#6850²²⁷²](#): Fix smartypants module calls re.sub() with wrong options
- [#6824²²⁷³](#): HTML search: If a search term is partially matched in the title and fully matched in a text paragraph on the same page, the search does not include this match.
- [#6848²²⁷⁴](#): config.py shouldn't pop extensions from overrides
- [#6867²²⁷⁵](#): text: extra spaces are inserted to hyphenated words on folding lines
- [#6886²²⁷⁶](#): LaTeX: xelatex converts straight double quotes into right curly ones (shows when `smartquotes` is False)
- [#6890²²⁷⁷](#): LaTeX: even with smartquotes off, PDF output transforms straight quotes and consecutive hyphens into curly quotes and dashes
- [#6876²²⁷⁸](#): LaTeX: multi-line display of authors on title page has ragged edges
- [#6887²²⁷⁹](#): Sphinx crashes with Docutils 0.16b0
- [#6920²²⁸⁰](#): sphinx-build: A console message is wrongly highlighted
- [#6900²²⁸¹](#): sphinx-build: -D option does not considers 0 and 1 as a boolean value

²²⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/6712>

²²⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/6741>

²²⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/6001>

²²⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/6804>

²²⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/6809>

²²⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/6793>

²²⁷¹ <https://github.com/sphinx-doc/sphinx/issues/6813>

²²⁷² <https://github.com/sphinx-doc/sphinx/issues/6850>

²²⁷³ <https://github.com/sphinx-doc/sphinx/issues/6824>

²²⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/6848>

²²⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/6867>

²²⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/6886>

²²⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/6890>

²²⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/6876>

²²⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/6887>

²²⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/6920>

²²⁸¹ <https://github.com/sphinx-doc/sphinx/issues/6900>

14.84 Release 2.2.2 (released Dec 03, 2019)

Modifiche incompatibili

- [#6803](#)²²⁸²: For security reason of python, parallel mode is disabled on macOS and Python3.8+

Bugs fixed

- [#6776](#)²²⁸³: LaTeX: 2019-10-01 LaTeX release breaks `sphinxcyrillic.sty`
- [#6815](#)²²⁸⁴: i18n: French, Hindi, Chinese, Japanese and Korean translation messages has been broken
- [#6803](#)²²⁸⁵: parallel build causes `AttributeError` on macOS and Python3.8

14.85 Release 2.2.1 (released Oct 26, 2019)

Bugs fixed

- [#6641](#)²²⁸⁶: LaTeX: Undefined control sequence `\sphinxmaketitle`
- [#6710](#)²²⁸⁷: LaTeX not well configured for Greek language as main language
- [#6759](#)²²⁸⁸: validation of html static paths and extra paths no longer throws an error if the paths are in different directories

14.86 Release 2.2.0 (released Aug 19, 2019)

Modifiche incompatibili

- apidoc: template files are renamed to `.rst_t`
- html: Field lists will be styled by grid layout

²²⁸² <https://github.com/sphinx-doc/sphinx/issues/6803>

²²⁸³ <https://github.com/sphinx-doc/sphinx/issues/6776>

²²⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/6815>

²²⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/6803>

²²⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/6641>

²²⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/6710>

²²⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/6759>

Deprecato

- `sphinx.domains.math.MathDomain.add_equation()`
- `sphinx.domains.math.MathDomain.get_next_equation_number()`
- Gli argomenti `info` e `warn` di `sphinx.ext.autosummary.generate.generate_autosummary_docs()`
- `sphinx.ext.autosummary.generate._simple_info()`
- `sphinx.ext.autosummary.generate._simple_warn()`
- `sphinx.ext.todo.merge_info()`
- `sphinx.ext.todo.process_todo_nodes()`
- `sphinx.ext.todo.process_todos()`
- `sphinx.ext.todo.purge_todos()`

Funzionalità aggiunte

- [#5124](https://github.com/sphinx-doc/sphinx/issues/5124)²²⁸⁹: `graphviz: :graphviz_dot:` option is renamed to `:layout:`
- [#1464](https://github.com/sphinx-doc/sphinx/issues/1464)²²⁹⁰: `html:` emit a warning if `html_static_path` and `html_extra_path` directories are inside output directory
- [#6514](https://github.com/sphinx-doc/sphinx/issues/6514)²²⁹¹: `html:` Add a label to search input for accessibility purposes
- [#5602](https://github.com/sphinx-doc/sphinx/issues/5602)²²⁹²: `apidoc:` Add `--templatedir` option
- [#6475](https://github.com/sphinx-doc/sphinx/issues/6475)²²⁹³: Add override argument to `app.add_autodocumenter()`
- [#6310](https://github.com/sphinx-doc/sphinx/issues/6310)²²⁹⁴: `imgmath:` let `imgmath_use_preview` work also with the SVG format for images rendering inline math
- [#6533](https://github.com/sphinx-doc/sphinx/issues/6533)²²⁹⁵: LaTeX: refactor `visit_enumerated_list()` to use `\sphinxsetlistlabels`
- [#6628](https://github.com/sphinx-doc/sphinx/issues/6628)²²⁹⁶: `quickstart:` Use `https://docs.python.org/3/` for default setting of `intersphinx_mapping`
- [#6419](https://github.com/sphinx-doc/sphinx/issues/6419)²²⁹⁷: `sphinx-build:` give reasons why rebuilt

²²⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/5124>

²²⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/1464>

²²⁹¹ <https://github.com/sphinx-doc/sphinx/issues/6514>

²²⁹² <https://github.com/sphinx-doc/sphinx/issues/5602>

²²⁹³ <https://github.com/sphinx-doc/sphinx/issues/6475>

²²⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/6310>

²²⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/6533>

²²⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/6628>

²²⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/6419>

Bugs fixed

- py domain: duplicated warning does not point the location of source code
- #6499²²⁹⁸: html: Sphinx never updates a copy of *html_logo* even if original file has changed
- #1125²²⁹⁹: html theme: scrollbar is hard to see on classic theme and macOS
- #5502²³⁰⁰: linkcheck: Consider HTTP 503 response as not an error
- #6439²³⁰¹: Make generated download links reproducible
- #6486²³⁰²: UnboundLocalError is raised if broken extension installed
- #6567²³⁰³: autodoc: *autodoc_inherit_docstrings* does not effect to `__init__()` and `__new__()`
- #6574²³⁰⁴: autodoc: *autodoc_member_order* does not refer order of imports when 'bysource' order
- #6574²³⁰⁵: autodoc: missing type annotation for variadic and keyword parameters
- #6589²³⁰⁶: autodoc: Formatting issues with `autodoc_typehints='none'`
- #6605²³⁰⁷: autodoc: crashed when target code contains custom method-like objects
- #6498²³⁰⁸: autosummary: crashed with wrong `autosummary_generate` setting
- #6507²³⁰⁹: autosummary: crashes without no `autosummary_generate` setting
- #6511²³¹⁰: LaTeX: autonumbered list can not be customized in LaTeX since Sphinx 1.8.0 (refs: #6533²³¹¹)
- #6531²³¹²: Failed to load last environment object when extension added
- #736²³¹³: Invalid sort in pair index
- #6527²³¹⁴: *last_updated* wrongly assumes timezone as UTC

²²⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/6499>

²²⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/1125>

²³⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/5502>

²³⁰¹ <https://github.com/sphinx-doc/sphinx/issues/6439>

²³⁰² <https://github.com/sphinx-doc/sphinx/issues/6486>

²³⁰³ <https://github.com/sphinx-doc/sphinx/issues/6567>

²³⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/6574>

²³⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/6574>

²³⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/6589>

²³⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/6605>

²³⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/6498>

²³⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/6507>

²³¹⁰ <https://github.com/sphinx-doc/sphinx/issues/6511>

²³¹¹ <https://github.com/sphinx-doc/sphinx/issues/6533>

²³¹² <https://github.com/sphinx-doc/sphinx/issues/6531>

²³¹³ <https://github.com/sphinx-doc/sphinx/issues/736>

²³¹⁴ <https://github.com/sphinx-doc/sphinx/issues/6527>

- [#5592](#)²³¹⁵: std domain: `option` directive registers an index entry for each comma separated option
- [#6549](#)²³¹⁶: sphinx-build: Escaped characters in error messages
- [#6545](#)²³¹⁷: doctest comments not getting trimmed since Sphinx 1.8.0
- [#6561](#)²³¹⁸: glossary: Wrong hyperlinks are generated for non alphanumeric terms
- [#6620](#)²³¹⁹: i18n: classifiers of definition list are not translated with Docutils 0.15
- [#6474](#)²³²⁰: DocFieldTransformer raises AttributeError when given directive is not a subclass of ObjectDescription

14.87 Release 2.1.2 (released Jun 19, 2019)

Bugs fixed

- [#6497](#)²³²¹: custom lexers fails highlighting when syntax error
- [#6478](#)²³²², [#6488](#)²³²³: info field lists are incorrectly recognized

14.88 Release 2.1.1 (released Jun 10, 2019)

Modifiche incompatibili

- [#6447](#)²³²⁴: autodoc: Stop to generate document for undocumented module variables

Bugs fixed

- [#6442](#)²³²⁵: LaTeX: admonitions of `note` type can get separated from immediately preceding section title by pagebreak
- [#6448](#)²³²⁶: autodoc: crashed when aut documenting classes with `__slots__ = None`
- [#6451](#)²³²⁷: autodoc: generates docs for "optional import"ed modules as variables

²³¹⁵ <https://github.com/sphinx-doc/sphinx/issues/5592>

²³¹⁶ <https://github.com/sphinx-doc/sphinx/issues/6549>

²³¹⁷ <https://github.com/sphinx-doc/sphinx/issues/6545>

²³¹⁸ <https://github.com/sphinx-doc/sphinx/issues/6561>

²³¹⁹ <https://github.com/sphinx-doc/sphinx/issues/6620>

²³²⁰ <https://github.com/sphinx-doc/sphinx/issues/6474>

²³²¹ <https://github.com/sphinx-doc/sphinx/issues/6497>

²³²² <https://github.com/sphinx-doc/sphinx/issues/6478>

²³²³ <https://github.com/sphinx-doc/sphinx/issues/6488>

²³²⁴ <https://github.com/sphinx-doc/sphinx/issues/6447>

²³²⁵ <https://github.com/sphinx-doc/sphinx/issues/6442>

²³²⁶ <https://github.com/sphinx-doc/sphinx/issues/6448>

²³²⁷ <https://github.com/sphinx-doc/sphinx/issues/6451>

- #6452²³²⁸: autosummary: crashed when generating document of properties
- #6455²³²⁹: napoleon: docstrings for properties are not processed
- #6436²³³⁰: napoleon: "Unknown target name" error if variable name ends with underscore
- #6440²³³¹: apidoc: missing blank lines between modules

14.89 Release 2.1.0 (released Jun 02, 2019)

Modifiche incompatibili

- Ignore filenames without file extension given to `Builder.build_specific()` API directly
- #6230²³³²: The anchor of term in glossary directive is changed if it is consisted by non-ASCII characters
- #4550²³³³: html: Centering tables by default using CSS
- #6239²³³⁴: latex: xelatex and xeCJK are used for Chinese documents by default
- `Sphinx.add_lexer()` now takes a `Lexer` class instead of instance. An instance of lexers are still supported until Sphinx 3.x.

Deprecato

- `sphinx.builders.latex.LaTeXBuilder.apply_transforms()`
- `sphinx.builders._epub_base.EpubBuilder.esc()`
- `sphinx.directives.Acks`
- `sphinx.directives.Author`
- `sphinx.directives.Centered`
- `sphinx.directives.Class`
- `sphinx.directives.CodeBlock`
- `sphinx.directives.Figure`
- `sphinx.directives.HList`
- `sphinx.directives.Highlight`

²³²⁸ <https://github.com/sphinx-doc/sphinx/issues/6452>

²³²⁹ <https://github.com/sphinx-doc/sphinx/issues/6455>

²³³⁰ <https://github.com/sphinx-doc/sphinx/issues/6436>

²³³¹ <https://github.com/sphinx-doc/sphinx/issues/6440>

²³³² <https://github.com/sphinx-doc/sphinx/issues/6230>

²³³³ <https://github.com/sphinx-doc/sphinx/issues/4550>

²³³⁴ <https://github.com/sphinx-doc/sphinx/issues/6239>

- `sphinx.directives.Include`
- `sphinx.directives.Index`
- `sphinx.directives.LiteralInclude`
- `sphinx.directives.Meta`
- `sphinx.directives.Only`
- `sphinx.directives.SeeAlso`
- `sphinx.directives.TabularColumns`
- `sphinx.directives.TocTree`
- `sphinx.directives.VersionChange`
- `sphinx.domains.python.PyClassmember`
- `sphinx.domains.python.PyModulelevel`
- `sphinx.domains.std.StandardDomain._resolve_citation_xref()`
- `sphinx.domains.std.StandardDomain.note_citations()`
- `sphinx.domains.std.StandardDomain.note_citation_refs()`
- `sphinx.domains.std.StandardDomain.note_labels()`
- `sphinx.environment.NoUri`
- `sphinx.ext.apidoc.format_directive()`
- `sphinx.ext.apidoc.format_heading()`
- `sphinx.ext.apidoc.makename()`
- `sphinx.ext.autodoc.importer.MockFinder`
- `sphinx.ext.autodoc.importer.MockLoader`
- `sphinx.ext.autodoc.importer.mock()`
- `sphinx.ext.autosummary.autolink_role()`
- `sphinx.ext.imgmath.DOC_BODY`
- `sphinx.ext.imgmath.DOC_BODY_PREVIEW`
- `sphinx.ext.imgmath.DOC_HEAD`
- `sphinx.transforms.CitationReferences`
- `sphinx.transforms.SmartQuotesSkipper`
- `sphinx.util.docfields.DocFieldTransformer.preprocess_fielddtypes()`
- `sphinx.util.node.find_source_node()`
- `sphinx.util.i18n.find_catalog()`
- `sphinx.util.i18n.find_catalog_files()`

- `sphinx.util.i18n.find_catalog_source_files()`

For more details, see *deprecation APIs list*.

Funzionalità aggiunte

- Add a helper class `sphinx.transforms.post_transforms.SphinxPostTransform`
- Add helper methods
 - `PythonDomain.note_module()`
 - `PythonDomain.note_object()`
 - `SphinxDirective.set_source_info()`
- #6180²³³⁵: Support `--keep-going` with `BuildDoc` setup command
- `math` directive now supports `:class:` option
- `todo`: `todo` directive now supports `:name:` option
- Enable override via environment of `SPHINXOPTS` and `SPHINXBUILD` Makefile variables (refs: #6232²³³⁶, #6303²³³⁷)
- #6287²³³⁸: `autodoc`: Unable to document bound instance methods exported as module functions
- #6289²³³⁹: `autodoc`: `autodoc_default_options` now supports `imported-members` option
- #4777²³⁴⁰: `autodoc`: Support coroutine
- #744²³⁴¹: `autodoc`: Support `abstractmethod`
- #6325²³⁴²: `autodoc`: Support attributes in `__slots__`. For dict-style `__slots__`, `autodoc` considers values as a docstring of the attribute
- #6361²³⁴³: `autodoc`: Add `autodoc_typehints` to suppress typehints from signature
- #1063²³⁴⁴: `autodoc`: `automodule` directive now handles undocumented module level variables
- #6212²³⁴⁵ `autosummary`: Add `autosummary_imported_members` to display imported members on `autosummary`

²³³⁵ <https://github.com/sphinx-doc/sphinx/issues/6180>

²³³⁶ <https://github.com/sphinx-doc/sphinx/issues/6232>

²³³⁷ <https://github.com/sphinx-doc/sphinx/issues/6303>

²³³⁸ <https://github.com/sphinx-doc/sphinx/issues/6287>

²³³⁹ <https://github.com/sphinx-doc/sphinx/issues/6289>

²³⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/4777>

²³⁴¹ <https://github.com/sphinx-doc/sphinx/issues/744>

²³⁴² <https://github.com/sphinx-doc/sphinx/issues/6325>

²³⁴³ <https://github.com/sphinx-doc/sphinx/issues/6361>

²³⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/1063>

²³⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/6212>

- [#6271](#)²³⁴⁶: `make clean` is catastrophically broken if building into `'.'`
- [#6363](#)²³⁴⁷: Support `%0%` environment variable in `make.bat`
- [#4777](#)²³⁴⁸: `py` domain: Add `:async:` option to `py:function` directive
- `py` domain: Add new options to `py:method` directive
 - `:abstractmethod:`
 - `:async:`
 - `:classmethod:`
 - `:property:`
 - `:staticmethod:`
- `rst` domain: Add `rst:directive:option` directive to describe the option for directive
- [#6306](#)²³⁴⁹: `html`: Add a label to search form for accessibility purposes
- [#4390](#)²³⁵⁰: `html`: Consistent and semantic CSS for signatures
- [#6358](#)²³⁵¹: The `rawsource` property of production nodes now contains the full production rule
- [#6373](#)²³⁵²: `autosectionlabel`: Allow suppression of warnings
- `coverage`: Support a new `coverage_ignore_pyobjects` option
- [#6239](#)²³⁵³: `latex`: Support to build Chinese documents

Bugs fixed

- [#6230](#)²³⁵⁴: Inappropriate `node_id` has been generated by `glossary` directive if term is consisted by non-ASCII characters
- [#6213](#)²³⁵⁵: `ifconfig`: contents after headings are not shown
- commented term in `glossary` directive is wrongly recognized
- [#6299](#)²³⁵⁶: `rst` domain: `rst:directive` directive generates waste space
- [#6379](#)²³⁵⁷: `py` domain: Module index (`py-modindex.html`) has duplicate titles

²³⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/6271>

²³⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/6363>

²³⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/4777>

²³⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/6306>

²³⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/4390>

²³⁵¹ <https://github.com/sphinx-doc/sphinx/issues/6358>

²³⁵² <https://github.com/sphinx-doc/sphinx/issues/6373>

²³⁵³ <https://github.com/sphinx-doc/sphinx/issues/6239>

²³⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/6230>

²³⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/6213>

²³⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/6299>

²³⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/6379>

- [#6331²³⁵⁸](#): man: invalid output when doctest follows rubric
- [#6351²³⁵⁹](#): "Hyperlink target is not referenced" message is shown even if referenced
- [#6165²³⁶⁰](#): autodoc: `tab_width` setting of Docutils has been ignored
- [#6347²³⁶¹](#): autodoc: crashes with a plain Tuple on Python 3.6 and 3.5
- [#6311²³⁶²](#): autosummary: autosummary table gets confused by complex type hints
- [#6350²³⁶³](#): autosummary: confused by an argument having some kind of default value
- Generated Makefiles lack a final EOL (refs: [#6232²³⁶⁴](#))
- [#6375²³⁶⁵](#): extlinks: Cannot escape angle brackets in link caption
- [#6378²³⁶⁶](#): linkcheck: Send commonly used User-Agent
- [#6387²³⁶⁷](#): html search: failed to search document with haiku and scrolls themes
- [#6408²³⁶⁸](#): html search: Fix the ranking of search results
- [#6406²³⁶⁹](#): Wrong year is returned for `SOURCE_DATE_EPOCH`
- [#6402²³⁷⁰](#): image directive crashes by unknown image format
- [#6286²³⁷¹](#): C++, allow 8 and 9 in hexadecimal integer literals.
- [#6305²³⁷²](#): Fix the string in quickstart for 'path' argument of parser
- LaTeX: Figures in admonitions produced errors (refs: [#6364²³⁷³](#))

14.90 Release 2.0.1 (released Apr 08, 2019)

Bugs fixed

- LaTeX: some system labels are not translated
- `RemovedInSphinx30Warning` is marked as pending
- deprecation warnings are not emitted

²³⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/6331>

²³⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/6351>

²³⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/6165>

²³⁶¹ <https://github.com/sphinx-doc/sphinx/issues/6347>

²³⁶² <https://github.com/sphinx-doc/sphinx/issues/6311>

²³⁶³ <https://github.com/sphinx-doc/sphinx/issues/6350>

²³⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/6232>

²³⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/6375>

²³⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/6378>

²³⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/6387>

²³⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/6408>

²³⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/6406>

²³⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/6402>

²³⁷¹ <https://github.com/sphinx-doc/sphinx/issues/6286>

²³⁷² <https://github.com/sphinx-doc/sphinx/issues/6305>

²³⁷³ <https://github.com/sphinx-doc/sphinx/issues/6364>

- `sphinx.application.CONFIG_FILENAME`
- `sphinx.builders.htmlhelp`
- `viewcode_import`
- #6208²³⁷⁴: C++, properly parse full xrefs that happen to have a short xref as prefix
- #6220²³⁷⁵, #6225²³⁷⁶: napoleon: AttributeError is raised for raised section having references
- #6245²³⁷⁷: circular import error on importing SerializingHTMLBuilder
- #6243²³⁷⁸: LaTeX: 'releasename' setting for latex_elements is ignored
- #6244²³⁷⁹: html: Search function is broken with 3rd party themes
- #6263²³⁸⁰: html: HTML5Translator crashed with invalid field node
- #6262²³⁸¹: html theme: The style of field lists has changed in bizstyle theme

14.91 Release 2.0.0 (released Mar 29, 2019)

Dipendenze

2.0.0b1

- LaTeX builder now depends on TeX Live 2015 or above.
- LaTeX builder (with 'pdf`latex`' `latex_engine`) will process Unicode Greek letters in text (not in math mark-up) via the text font and will not escape them to math mark-up. See the discussion of the 'fontenc' key of `latex_elements`; such (optional) support for Greek adds, for example on Ubuntu xenial, the `texlive-lang-greek` and (if default font set-up is not modified) `cm-super(-minimal)` as additional Sphinx LaTeX requirements.
- LaTeX builder with `latex_engine` set to 'xelatex' or to 'lualatex' requires (by default) the FreeFont fonts, which in Ubuntu xenial are provided by package `fonts-freefont-otf`, and e.g. in Fedora 29 via package `texlive-gnu-freefont`.
- requests 2.5.0 or above
- The six package is no longer a dependency
- The sphinxcontrib-websupport package is no longer a dependency
- Some packages are separated to sub packages:

²³⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/6208>

²³⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/6220>

²³⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/6225>

²³⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/6245>

²³⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/6243>

²³⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/6244>

²³⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/6263>

²³⁸¹ <https://github.com/sphinx-doc/sphinx/issues/6262>

- sphinxcontrib.applehelp
- sphinxcontrib.devhelp
- sphinxcontrib.htmlhelp
- sphinxcontrib.jsmath
- sphinxcontrib.serializinghtml
- sphinxcontrib.qthelp

Modifiche incompatibili

2.0.0b1

- Drop python 2.7 and 3.4 support
- Drop Docutils 0.11 support
- Drop features and APIs deprecated in 1.7.x
- The default setting for `master_doc` is changed to 'index' which has been longly used as default of sphinx-quickstart.
- LaTeX: Move message resources to `sphinxmessage.sty`
- LaTeX: Stop using `\captions<lang>` macro for some labels
- LaTeX: for 'xelatex' and 'lualatex', use the FreeFont OpenType fonts as default choice (refs: [#5645](#)²³⁸²)
- LaTeX: 'xelatex' and 'lualatex' now use `\small` in code-blocks (due to FreeMono character width) like 'pdflatex' already did (due to Courier character width). You may need to adjust this via `latex_elements` 'fvset' key, in case of usage of some other OpenType fonts (refs: [#5768](#)²³⁸³)
- LaTeX: Greek letters in text are not escaped to math mode mark-up, and they will use the text font not the math font. The LGR font encoding must be added to the 'fontenc' key of `latex_elements` for this to work (only if it is needed by the document, of course).
- LaTeX: setting the `language` to 'en' triggered Sonny option of `fncychap`, now it is Bjarne to match case of no language specified. (refs: [#5772](#)²³⁸⁴)
- [#5770](#)²³⁸⁵: doctest: Follow `highlight_language` on highlighting doctest block. As a result, they are highlighted as python3 by default.
- The order of argument for `HTMLTranslator`, `HTML5Translator` and `ManualPageTranslator` are changed

²³⁸² <https://github.com/sphinx-doc/sphinx/issues/5645>

²³⁸³ <https://github.com/sphinx-doc/sphinx/issues/5768>

²³⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/5772>

²³⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/5770>

- LaTeX: hard-coded redefinitions of `\l@section` and `\l@subsection` formerly done during loading of 'manual' docclass get executed later, at time of `\sphinxtableofcontents`. This means that custom user definitions from LaTeX preamble now get overwritten. Use `\sphinxtableofcontentshook` to insert custom user definitions. See *Le Macro*.
- quickstart: Simplify generated `conf.py`
- [#4148](#)²³⁸⁶: quickstart: some questions are removed. They are still able to specify via command line options
- websupport: unbundled from Sphinx core. Please use `sphinxcontrib-websupport`
- C++, the visibility of base classes is now always rendered as present in the input. That is, `private` is now shown, where it was ellided before.
- LaTeX: graphics inclusion of oversized images rescales to not exceed the text width and height, even if width and/or height option were used. (refs: [#5956](#)²³⁸⁷)
- epub: `epub_title` defaults to the *project* option
- [#4550](#)²³⁸⁸: All tables and figures without `align` option are displayed to center
- [#4587](#)²³⁸⁹: html: Output HTML5 by default

2.0.0b2

- texinfo: image files are copied into `name-figure` directory

Deprecato

2.0.0b1

- Support for evaluating Python 2 syntax is deprecated. This includes configuration files which should be converted to Python 3.
- The arguments of `EpubBuilder.build_mimetype()`, `EpubBuilder.build_container()`, `EpubBuilder.bulid_content()`, `EpubBuilder.build_toc()` and `EpubBuilder.build_epub()`
- The arguments of `Epub3Builder.build_navigation_doc()`
- The config variables
 - *html_experimental_html5_writer*
- The encoding argument of `autodoc.Documenter.get_doc()`, `autodoc.DocstringSignatureMixin.get_doc()`, `autodoc.DocstringSignatureMixin._find_signature()`, and `autodoc.ClassDocumenter.get_doc()` are deprecated.
- L'argomento `importer` di `sphinx.ext.autodoc.importer._MockModule`

²³⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/4148>

²³⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/5956>

²³⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/4550>

²³⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/4587>

- The `nodetype` argument of `sphinx.search.WordCollector.is_meta_keywords()`
- The `suffix` argument of `env.doc2path()` is deprecated.
- The `string style base` argument of `env.doc2path()` is deprecated.
- The fallback to allow omitting the `filename` argument from an overridden `IndexBuilder.feed()` method is deprecated.
- `sphinx.addnodes.abbreviation`
- `sphinx.application.Sphinx._setting_up_extension`
- `sphinx.builders.epub3.Epub3Builder.validate_config_value()`
- `sphinx.builders.html.SingleFileHTMLBuilder`
- `sphinx.builders.htmlhelp.HTMLHelpBuilder.open_file()`
- `sphinx.cmd.quickstart.term_decode()`
- `sphinx.cmd.quickstart.TERM_ENCODING`
- `sphinx.config.check_unicode()`
- `sphinx.config.string_classes`
- `sphinx.domains.cpp.DefinitionError.description`
- `sphinx.domains.cpp.NoOldIdError.description`
- `sphinx.domains.cpp.UnsupportedMultiCharacterCharLiteral.decoded`
- `sphinx.ext.autodoc.importer._MockImporter`
- `sphinx.ext.autosummary.Autosummary.warn()`
- `sphinx.ext.autosummary.Autosummary.genopt`
- `sphinx.ext.autosummary.Autosummary.warnings`
- `sphinx.ext.autosummary.Autosummary.result`
- `sphinx.ext.doctest.doctest_encode()`
- `sphinx.io.SphinxBaseFileInput`
- `sphinx.io.SphinxFileInput.supported`
- `sphinx.io.SphinxRSTFileInput`
- `sphinx.registry.SphinxComponentRegistry.add_source_input()`
- `sphinx.roles.abbr_role()`
- `sphinx.roles.emph_literal_role()`
- `sphinx.roles.menusel_role()`
- `sphinx.roles.index_role()`
- `sphinx.roles.indexmarkup_role()`

- `sphinx.testing.util.remove_unicode_literal()`
- `sphinx.util.attrdict`
- `sphinx.util.force_decode()`
- `sphinx.util.get_matching_docs()`
- `sphinx.util.inspect.Parameter`
- `sphinx.util.jsonimpl`
- `sphinx.util.osutil.EEXIST`
- `sphinx.util.osutil.EINVAL`
- `sphinx.util.osutil.ENOENT`
- `sphinx.util.osutil.EPIPE`
- `sphinx.util.osutil.walk()`
- `sphinx.util.PeekableIterator`
- `sphinx.util.pycompat.NoneType`
- `sphinx.util.pycompat.TextIOWrapper`
- `sphinx.util.pycompat.UnicodeMixin`
- `sphinx.util.pycompat.htmlescape`
- `sphinx.util.pycompat.indent`
- `sphinx.util.pycompat.sys_encoding`
- `sphinx.util.pycompat.terminal_safe()`
- `sphinx.util.pycompat.u`
- `sphinx.writers.latex.ExtBabel`
- `sphinx.writers.latex.LaTeXTranslator._make_visit_admonition()`
- `sphinx.writers.latex.LaTeXTranslator.babel_defmacro()`
- `sphinx.writers.latex.LaTeXTranslator.collect_footnotes()`
- `sphinx.writers.latex.LaTeXTranslator.generate_numfig_format()`
- `sphinx.writers.texinfo.TexinfoTranslator._make_visit_admonition()`
- `sphinx.writers.text.TextTranslator._make_depart_admonition()`
- template variables for LaTeX template
 - `logo`
 - `numfig_format`
 - `pageautorefname`
 - `translatablestrings`

For more details, see *deprecation APIs list*.

Funzionalità aggiunte

2.0.0b1

- [#1618](#)²³⁹⁰: The search results preview of generated HTML documentation is reader-friendlier: instead of showing the snippets as raw reStructuredText markup, Sphinx now renders the corresponding HTML. This means the Sphinx extension [Sphinx: pretty search results](#)²³⁹¹ is no longer necessary. Note that changes to the search function of your custom or 3rd-party HTML template might overwrite this improvement.
- [#4182](#)²³⁹²: autodoc: Support [suppress_warnings](#)
- [#5533](#)²³⁹³: autodoc: [autodoc_default_options](#) supports member-order
- [#5394](#)²³⁹⁴: autodoc: Display readable names in type annotations for mocked objects
- [#5459](#)²³⁹⁵: autodoc: [autodoc_default_options](#) accepts True as a value
- [#1148](#)²³⁹⁶: autodoc: Add [autodecorator](#) directive for decorators
- [#5635](#)²³⁹⁷: autosummary: Add [autosummary_mock_imports](#) to mock external libraries on importing targets
- [#4018](#)²³⁹⁸: htmlhelp: Add [htmlhelp_file_suffix](#) and [htmlhelp_link_suffix](#)
- [#5559](#)²³⁹⁹: text: Support complex tables (colspan and rowspan)
- LaTeX: support rendering (not in math, yet) of Greek and Cyrillic Unicode letters in non-Cyrillic document even with 'pdflatex' as [latex_engine](#) (refs: [#5645](#)²⁴⁰⁰)
- [#5660](#)²⁴⁰¹: The `versionadded`, `versionchanged` and `deprecated` directives are now generated with their own specific CSS classes (`added`, `changed` and `deprecated`, respectively) in addition to the generic `versionmodified` class.
- [#5841](#)²⁴⁰²: apidoc: Add `--extensions` option to `sphinx-apidoc`
- [#4981](#)²⁴⁰³: C++, added an alias directive for inserting lists of declarations, that references existing declarations (e.g., for making a synopsis).
- C++: add `cpp:struct` to complement `cpp:class`.

²³⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/1618>

²³⁹¹ <https://github.com/sphinx-contrib/sphinx-pretty-searchresults>

²³⁹² <https://github.com/sphinx-doc/sphinx/issues/4182>

²³⁹³ <https://github.com/sphinx-doc/sphinx/issues/5533>

²³⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/5394>

²³⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/5459>

²³⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/1148>

²³⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/5635>

²³⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/4018>

²³⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/5559>

²⁴⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/5645>

²⁴⁰¹ <https://github.com/sphinx-doc/sphinx/issues/5660>

²⁴⁰² <https://github.com/sphinx-doc/sphinx/issues/5841>

²⁴⁰³ <https://github.com/sphinx-doc/sphinx/issues/4981>

- #1341²⁴⁰⁴ the HTML search considers words that contain a search term of length three or longer a match.
- #4611²⁴⁰⁵: epub: Show warning for duplicated ToC entries
- #1851²⁴⁰⁶: Allow to omit an argument for `code-block` directive. If omitted, it follows `highlight` or `highlight_language`
- #4587²⁴⁰⁷: html: Add `html4_writer` to use old HTML4 writer
- #6016²⁴⁰⁸: HTML search: A placeholder for the search summary prevents search result links from changing their position when the search terminates. This makes navigating search results easier.
- #5196²⁴⁰⁹: linkcheck also checks remote images exist
- #5924²⁴¹⁰: githubpages: create CNAME file for custom domains when `html_baseurl` set
- #4261²⁴¹¹: autosectionlabel: restrict the labeled sections by new config value; `autosectionlabel_maxdepth`

Bugs fixed

2.0.0b1

- #1682²⁴¹²: LaTeX: writer should not translate Greek unicode, but use textgreek package
- #5247²⁴¹³: LaTeX: PDF does not build with default font config for Russian language and 'xelatex' or 'lualatex' as `latex_engine` (refs: #5251²⁴¹⁴)
- #5248²⁴¹⁵: LaTeX: Greek letters in section titles disappear from PDF bookmarks
- #5249²⁴¹⁶: LaTeX: Unicode Greek letters in math directive break PDF build (fix requires extra set-up, see `latex_elements` 'textgreek' key and/or `latex_engine` setting)
- #5772²⁴¹⁷: LaTeX: should the Bjarne style of fncychap be used for English also if passed as language option?
- #5179²⁴¹⁸: LaTeX: (lualatex only) escaping of > by `\textgreater{}` is not enough as `\textgreater{}\textgreater{}` applies TeX-ligature

²⁴⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/1341>

²⁴⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/4611>

²⁴⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/1851>

²⁴⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/4587>

²⁴⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/6016>

²⁴⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/5196>

²⁴¹⁰ <https://github.com/sphinx-doc/sphinx/issues/5924>

²⁴¹¹ <https://github.com/sphinx-doc/sphinx/issues/4261>

²⁴¹² <https://github.com/sphinx-doc/sphinx/issues/1682>

²⁴¹³ <https://github.com/sphinx-doc/sphinx/issues/5247>

²⁴¹⁴ <https://github.com/sphinx-doc/sphinx/issues/5251>

²⁴¹⁵ <https://github.com/sphinx-doc/sphinx/issues/5248>

²⁴¹⁶ <https://github.com/sphinx-doc/sphinx/issues/5249>

²⁴¹⁷ <https://github.com/sphinx-doc/sphinx/issues/5772>

²⁴¹⁸ <https://github.com/sphinx-doc/sphinx/issues/5179>

- LaTeX: project name is not escaped if `latex_documents` omitted
- LaTeX: authors are not shown if `latex_documents` omitted
- HTML: Invalid HTML5 file is generated for a glossary having multiple terms for one description (refs: #4611²⁴¹⁹)
- QtHelp: OS dependent path separator is used in .qhp file
- HTML search: search always returns nothing when multiple search terms are used and one term is shorter than three characters

2.0.0b2

- #6096²⁴²⁰: html: Anchor links are not added to figures
- #3620²⁴²¹: html: Defer searchindex.js rather than loading it via ajax
- #6113²⁴²²: html: Table cells and list items have large margins
- #5508²⁴²³: linenothreshold option for highlight directive was ignored
- texinfo: make install-info causes syntax error
- texinfo: make install-info fails on macOS
- #3079²⁴²⁴: texinfo: image files are not copied on make install-info
- #5391²⁴²⁵: A cross reference in heading is rendered as literal
- #5946²⁴²⁶: C++, fix cpp:alias problems in LaTeX (and singlehtml)
- #6147²⁴²⁷: classes attribute of citation_reference node is lost
- AssertionError is raised when custom citation_reference node having classes attribute refers missing citation (refs: #6147²⁴²⁸)
- #2155²⁴²⁹: Support code directive
- C++, fix parsing of braced initializers.
- #6172²⁴³⁰: AttributeError is raised for old styled index nodes
- #4872²⁴³¹: inheritance_diagram: correctly describe behavior of parts option in docs, allow negative values.

²⁴¹⁹ <https://github.com/sphinx-doc/sphinx/issues/4611>

²⁴²⁰ <https://github.com/sphinx-doc/sphinx/issues/6096>

²⁴²¹ <https://github.com/sphinx-doc/sphinx/issues/3620>

²⁴²² <https://github.com/sphinx-doc/sphinx/issues/6113>

²⁴²³ <https://github.com/sphinx-doc/sphinx/issues/5508>

²⁴²⁴ <https://github.com/sphinx-doc/sphinx/issues/3079>

²⁴²⁵ <https://github.com/sphinx-doc/sphinx/issues/5391>

²⁴²⁶ <https://github.com/sphinx-doc/sphinx/issues/5946>

²⁴²⁷ <https://github.com/sphinx-doc/sphinx/issues/6147>

²⁴²⁸ <https://github.com/sphinx-doc/sphinx/issues/6147>

²⁴²⁹ <https://github.com/sphinx-doc/sphinx/issues/2155>

²⁴³⁰ <https://github.com/sphinx-doc/sphinx/issues/6172>

²⁴³¹ <https://github.com/sphinx-doc/sphinx/issues/4872>

- [#6178](#)²⁴³²: i18n: Captions missing in translations for hidden TOCs

2.0.0 final

- [#6196](#)²⁴³³: py domain: unexpected prefix is generated

Testing

2.0.0b1

- Stop to use SPHINX_TEST_TEMPDIR envvar

2.0.0b2

- Add a helper function: `sphinx.testing.restructuredtext.parse()`

14.92 Release 1.8.6 (released Nov 18, 2021)

Dipendenze

- [#9807](#)²⁴³⁴: Restrict Docutils to 0.17.x or older

14.93 Release 1.8.5 (released Mar 10, 2019)

Bugs fixed

- LaTeX: Remove extraneous space after author names on PDF title page (refs: [#6004](#)²⁴³⁵)
- [#6026](#)²⁴³⁶: LaTeX: A cross reference to definition list does not work
- [#6046](#)²⁴³⁷: LaTeX: `TypeError` is raised when invalid `latex_elements` given
- [#6067](#)²⁴³⁸: LaTeX: images having a target are concatenated to next line
- [#6067](#)²⁴³⁹: LaTeX: images having a target are not aligned even if specified
- [#6149](#)²⁴⁴⁰: LaTeX: `:index:` role in titles causes Use of `\@icentercr` doesn't match its definition error on `latexpdf` build
- [#6019](#)²⁴⁴¹: `imgconverter`: Including multipage PDF fails

²⁴³² <https://github.com/sphinx-doc/sphinx/issues/6178>

²⁴³³ <https://github.com/sphinx-doc/sphinx/issues/6196>

²⁴³⁴ <https://github.com/sphinx-doc/sphinx/issues/9807>

²⁴³⁵ <https://github.com/sphinx-doc/sphinx/issues/6004>

²⁴³⁶ <https://github.com/sphinx-doc/sphinx/issues/6026>

²⁴³⁷ <https://github.com/sphinx-doc/sphinx/issues/6046>

²⁴³⁸ <https://github.com/sphinx-doc/sphinx/issues/6067>

²⁴³⁹ <https://github.com/sphinx-doc/sphinx/issues/6067>

²⁴⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/6149>

²⁴⁴¹ <https://github.com/sphinx-doc/sphinx/issues/6019>

- [#6047](#)²⁴⁴²: autodoc: autofunction emits a warning for method objects
- [#6028](#)²⁴⁴³: graphviz: Ensure the graphviz filenames are reproducible
- [#6068](#)²⁴⁴⁴: doctest: skipif option may remove the code block from documentation
- [#6136](#)²⁴⁴⁵: :name: option for math directive causes a crash
- [#6139](#)²⁴⁴⁶: intersphinx: ValueError on failure reporting
- [#6135](#)²⁴⁴⁷: changes: Fix UnboundLocalError when any module found
- [#3859](#)²⁴⁴⁸: manpage: code-block captions are not displayed correctly

14.94 Release 1.8.4 (released Feb 03, 2019)

Bugs fixed

- [#3707](#)²⁴⁴⁹: latex: no bold checkmark (✓) available.
- [#5605](#)²⁴⁵⁰: with the documentation language set to Chinese, English words could not be searched.
- [#5889](#)²⁴⁵¹: LaTeX: user numfig_format is stripped of spaces and may cause build failure
- C++, fix hyperlinks for declarations involving east cv-qualifiers.
- [#5755](#)²⁴⁵²: C++, fix duplicate declaration error on function templates with constraints in the return type.
- C++, parse unary right fold expressions and binary fold expressions.
- pycode could not handle egg files on windows
- [#5928](#)²⁴⁵³: KeyError: 'DOCUTILSCONFIG' when running build
- [#5936](#)²⁴⁵⁴: LaTeX: PDF build broken by inclusion of image taller than page height in an admonition
- [#5231](#)²⁴⁵⁵: "make html" does not read and build "po" files in "locale" dir

²⁴⁴² <https://github.com/sphinx-doc/sphinx/issues/6047>

²⁴⁴³ <https://github.com/sphinx-doc/sphinx/issues/6028>

²⁴⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/6068>

²⁴⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/6136>

²⁴⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/6139>

²⁴⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/6135>

²⁴⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/3859>

²⁴⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/3707>

²⁴⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/5605>

²⁴⁵¹ <https://github.com/sphinx-doc/sphinx/issues/5889>

²⁴⁵² <https://github.com/sphinx-doc/sphinx/issues/5755>

²⁴⁵³ <https://github.com/sphinx-doc/sphinx/issues/5928>

²⁴⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/5936>

²⁴⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/5231>

- #5954²⁴⁵⁶: `:scale:` image option may break PDF build if image in an admonition
- #5966²⁴⁵⁷: mathjax has not been loaded on incremental build
- #5960²⁴⁵⁸: LaTeX: modified PDF layout since September 2018 TeXLive update of `parskip.sty`
- #5948²⁴⁵⁹: LaTeX: duplicated labels are generated for sections
- #5958²⁴⁶⁰: `versionadded` directive causes crash with Python 3.5.0
- #5995²⁴⁶¹: autodoc: `autodoc_mock_imports` conflict with metaclass on Python 3.7
- #5871²⁴⁶²: texinfo: a section title `.` is not allowed

14.95 Release 1.8.3 (released Dec 26, 2018)

Funzionalità aggiunte

- LaTeX: it is possible to insert custom material to appear on back of title page, see discussion of `'maketitle'` key of *latex_elements* (`'manual'` docclass only)

Bugs fixed

- #5725²⁴⁶³: mathjax: Use CDN URL for "latest" version by default
- #5460²⁴⁶⁴: html search does not work with some 3rd party themes
- #5520²⁴⁶⁵: LaTeX, caption package incompatibility since Sphinx 1.6
- #5614²⁴⁶⁶: autodoc: incremental build is broken when builtin modules are imported
- #5627²⁴⁶⁷: qthelp: `index.html` missing in QtHelp
- #5659²⁴⁶⁸: linkcheck: crashes for a hyperlink containing multibyte character
- #5754²⁴⁶⁹: DOC: Fix some mistakes in *Personalizzazione di LaTeX*
- #5810²⁴⁷⁰: LaTeX: `sphinxVerbatim` requires explicit `"hllines"` set-up since 1.6.6 (refs:

²⁴⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/5954>

²⁴⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/5966>

²⁴⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/5960>

²⁴⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/5948>

²⁴⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/5958>

²⁴⁶¹ <https://github.com/sphinx-doc/sphinx/issues/5995>

²⁴⁶² <https://github.com/sphinx-doc/sphinx/issues/5871>

²⁴⁶³ <https://github.com/sphinx-doc/sphinx/issues/5725>

²⁴⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/5460>

²⁴⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/5520>

²⁴⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/5614>

²⁴⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/5627>

²⁴⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/5659>

²⁴⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/5754>

²⁴⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/5810>

#1238²⁴⁷¹)

- #5636²⁴⁷²: C++, fix parsing of floating point literals.
- #5496²⁴⁷³ (again): C++, fix assertion in partial builds with duplicates.
- #5724²⁴⁷⁴: quickstart: sphinx-quickstart fails when \$LC_ALL is empty
- #1956²⁴⁷⁵: Default conf.py is not PEP8-compliant
- #5849²⁴⁷⁶: LaTeX: document class \maketitle is overwritten with no possibility to use original meaning in place of Sphinx custom one
- #5834²⁴⁷⁷: apidoc: wrong help for --tocfile
- #5800²⁴⁷⁸: todo: crashed if todo is defined in TextElement
- #5846²⁴⁷⁹: htmlhelp: convert hex escaping to decimal escaping in .hhc/.hhk files
- htmlhelp: broken .hhk file generated when title contains a double quote

14.96 Release 1.8.2 (released Nov 11, 2018)

Modifiche incompatibili

- #5497²⁴⁸⁰: Do not include MathJax.js and jsmath.js unless it is really needed

Funzionalità aggiunte

- #5471²⁴⁸¹: Show appropriate deprecation warnings

²⁴⁷¹ <https://github.com/sphinx-doc/sphinx/issues/1238>

²⁴⁷² <https://github.com/sphinx-doc/sphinx/issues/5636>

²⁴⁷³ <https://github.com/sphinx-doc/sphinx/issues/5496>

²⁴⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/5724>

²⁴⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/1956>

²⁴⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/5849>

²⁴⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/5834>

²⁴⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/5800>

²⁴⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/5846>

²⁴⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/5497>

²⁴⁸¹ <https://github.com/sphinx-doc/sphinx/issues/5471>

Bugs fixed

- #5490²⁴⁸²: latex: enumerated list causes a crash with recommonmark
- #5492²⁴⁸³: sphinx-build fails to build docs w/ Python < 3.5.2
- #3704²⁴⁸⁴: latex: wrong \label positioning for figures with a legend
- #5496²⁴⁸⁵: C++, fix assertion when a symbol is declared more than twice.
- #5493²⁴⁸⁶: gettext: crashed with broken template
- #5495²⁴⁸⁷: csv-table directive with file option in included file is broken (refs: #4821²⁴⁸⁸)
- #5498²⁴⁸⁹: autodoc: unable to find type hints for a `functools.partial`
- #5480²⁴⁹⁰: autodoc: unable to find type hints for unresolvable Forward references
- #5419²⁴⁹¹: incompatible math_block node has been generated
- #5548²⁴⁹²: Fix ensuredir() in case of pre-existing file
- #5549²⁴⁹³: graphviz Correctly deal with non-existing static dir
- #3002²⁴⁹⁴: i18n: multiple footnote_references referring same footnote cause duplicated node_ids
- #5563²⁴⁹⁵: latex: footnote_references generated by extension causes a LaTeX builder crash
- #5561²⁴⁹⁶: make all-pdf fails with old xindy version
- #5557²⁴⁹⁷: quickstart: --no-batchfile isn't honored
- #3080²⁴⁹⁸: texinfo: multiline rubrics are broken
- #3080²⁴⁹⁹: texinfo: multiline citations are broken

²⁴⁸² <https://github.com/sphinx-doc/sphinx/issues/5490>

²⁴⁸³ <https://github.com/sphinx-doc/sphinx/issues/5492>

²⁴⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/3704>

²⁴⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/5496>

²⁴⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/5493>

²⁴⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/5495>

²⁴⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/4821>

²⁴⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/5498>

²⁴⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/5480>

²⁴⁹¹ <https://github.com/sphinx-doc/sphinx/issues/5419>

²⁴⁹² <https://github.com/sphinx-doc/sphinx/issues/5548>

²⁴⁹³ <https://github.com/sphinx-doc/sphinx/issues/5549>

²⁴⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/3002>

²⁴⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/5563>

²⁴⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/5561>

²⁴⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/5557>

²⁴⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/3080>

²⁴⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/3080>

14.97 Release 1.8.1 (released Sep 22, 2018)

Modifiche incompatibili

- LaTeX `\pagestyle` commands have been moved to the LaTeX template. No changes in PDF, except possibly if `\sphinxtableofcontents`, which contained them, had been customized in `conf.py`. (refs: [#5455](#)²⁵⁰⁰)

Bugs fixed

- [#5418](#)²⁵⁰¹: Incorrect default path for sphinx-build -d/doctrees files
- [#5421](#)²⁵⁰²: autodoc emits deprecation warning for `autodoc_default_flags`
- [#5422](#)²⁵⁰³: lambda object causes PicklingError on storing environment
- [#5417](#)²⁵⁰⁴: Sphinx fails to build with syntax error in Python 2.7.5
- [#4911](#)²⁵⁰⁵: add latexpdf to make.bat for non make-mode
- [#5436](#)²⁵⁰⁶: Autodoc does not work with enum subclasses with properties/methods
- [#5437](#)²⁵⁰⁷: autodoc: crashed on modules importing eggs
- [#5433](#)²⁵⁰⁸: latex: ImportError: cannot import name 'DEFAULT_SETTINGS'
- [#5431](#)²⁵⁰⁹: autodoc: `autofunction` emits a warning for callable objects
- [#5457](#)²⁵¹⁰: Fix TypeError in error message when override is prohibited
- [#5453](#)²⁵¹¹: PDF builds of 'howto' documents have no page numbers
- [#5463](#)²⁵¹²: mathbase: `math_role` and `MathDirective` was disappeared in 1.8.0
- [#5454](#)²⁵¹³: latex: Index has disappeared from PDF for Japanese documents
- [#5432](#)²⁵¹⁴: py domain: `:type:` field can't process `:term:` references
- [#5426](#)²⁵¹⁵: py domain: TypeError has been raised for class attribute

²⁵⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/5455>

²⁵⁰¹ <https://github.com/sphinx-doc/sphinx/issues/5418>

²⁵⁰² <https://github.com/sphinx-doc/sphinx/issues/5421>

²⁵⁰³ <https://github.com/sphinx-doc/sphinx/issues/5422>

²⁵⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/5417>

²⁵⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/4911>

²⁵⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/5436>

²⁵⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/5437>

²⁵⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/5433>

²⁵⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/5431>

²⁵¹⁰ <https://github.com/sphinx-doc/sphinx/issues/5457>

²⁵¹¹ <https://github.com/sphinx-doc/sphinx/issues/5453>

²⁵¹² <https://github.com/sphinx-doc/sphinx/issues/5463>

²⁵¹³ <https://github.com/sphinx-doc/sphinx/issues/5454>

²⁵¹⁴ <https://github.com/sphinx-doc/sphinx/issues/5432>

²⁵¹⁵ <https://github.com/sphinx-doc/sphinx/issues/5426>

14.98 Release 1.8.0 (released Sep 13, 2018)

Dipendenze

1.8.0b1

- LaTeX: `latex_use_xindy`, if True (default for xelatex/lualatex), instructs make latexpdf to use **xindy** for general index. Make sure your LaTeX distribution includes it. (refs: [#5134](#)²⁵¹⁶)
- LaTeX: latexmk is required for make latexpdf on Windows

Modifiche incompatibili

1.8.0b2

- [#5282](#)²⁵¹⁷: html theme: refer `pygments_style` settings of HTML themes preferentially
- The URL of download files are changed
- [#5127](#)²⁵¹⁸: quickstart: Makefile and make.bat are not overwritten if exists

1.8.0b1

- [#5156](#)²⁵¹⁹: the `sphinx.ext.graphviz` extension runs `dot` in the directory of the document being built instead of in the root directory of the documentation.
- [#4460](#)²⁵²⁰: extensions which stores any data to environment should return the version of its env data structure as metadata. In detail, please see *Metadati dell'estensione*.
- Sphinx expects source parser modules to have supported file formats as `Parser.supported` attribute
- The default value of `epub_author` and `epub_publisher` are changed from 'unknown' to the value of `author`. This is same as a `conf.py` file sphinx-build generates.
- The `gettext_compact` attribute is removed from `document.settings` object. Please use `config.gettext_compact` instead.
- The processing order on reading phase is changed. `smart_quotes`, `sphinx domains`, `doctree-read` event and versioning doctrees are invoked earlier than so far. For more details, please read a description of `Sphinx.add_transform()`
- [#4827](#)²⁵²¹: All `substitution_definition` nodes are removed from doctree on reading phase
- `docutils.conf` in `$HOME` or `/etc` directories are ignored. Only `docutils.conf` from `confdir` is obeyed.

²⁵¹⁶ <https://github.com/sphinx-doc/sphinx/issues/5134>

²⁵¹⁷ <https://github.com/sphinx-doc/sphinx/issues/5282>

²⁵¹⁸ <https://github.com/sphinx-doc/sphinx/issues/5127>

²⁵¹⁹ <https://github.com/sphinx-doc/sphinx/issues/5156>

²⁵²⁰ <https://github.com/sphinx-doc/sphinx/issues/4460>

²⁵²¹ <https://github.com/sphinx-doc/sphinx/issues/4827>

- #789²⁵²²: `:samp:` role supports to escape curly braces with backslash
- #4811²⁵²³: The files under `html_static_path` are excluded from source files.
- latex: Use `\sphinxcite` for citation references instead `\hyperref`
- The config value `viewcode_import` is renamed to `viewcode_follow_imported_members` (refs: #4035²⁵²⁴)
- #1857²⁵²⁵: latex: `latex_show_pagerefs` does not add pagerefs for citations
- #4648²⁵²⁶: latex: Now "rubric" elements are rendered as unnumbered section title
- #4983²⁵²⁷: html: The anchor for productionlist tokens has been changed
- Modifying a template variable `script_files` in templates is allowed now. Please use `app.add_js_file()` instead.
- #5072²⁵²⁸: Save environment object also with only new documents
- #5035²⁵²⁹: qthelp builder allows dashes in `qthelp_namespace`
- LaTeX: with `lualatex` or `xelatex` use by default **xindy** as UTF-8 able replacement of **makeindex** (refs: #5134²⁵³⁰). After upgrading Sphinx, please clean latex build repertory of existing project before new build.
- #5163²⁵³¹: html: hlist items are now aligned to top
- `highlightlang` directive is processed on resolving phase
- #4000²⁵³²: LaTeX: template changed. Following elements moved to it:
 - `\begin{document}`
 - `shorthandoff` variable
 - `maketitle` variable
 - `tableofcontents` variable

²⁵²² <https://github.com/sphinx-doc/sphinx/issues/789>

²⁵²³ <https://github.com/sphinx-doc/sphinx/issues/4811>

²⁵²⁴ <https://github.com/sphinx-doc/sphinx/issues/4035>

²⁵²⁵ <https://github.com/sphinx-doc/sphinx/issues/1857>

²⁵²⁶ <https://github.com/sphinx-doc/sphinx/issues/4648>

²⁵²⁷ <https://github.com/sphinx-doc/sphinx/issues/4983>

²⁵²⁸ <https://github.com/sphinx-doc/sphinx/issues/5072>

²⁵²⁹ <https://github.com/sphinx-doc/sphinx/issues/5035>

²⁵³⁰ <https://github.com/sphinx-doc/sphinx/issues/5134>

²⁵³¹ <https://github.com/sphinx-doc/sphinx/issues/5163>

²⁵³² <https://github.com/sphinx-doc/sphinx/issues/4000>

Deprecato

1.8.0b2

- `sphinx.io.SphinxI18nReader.set_lineno_for_reporter()` is deprecated
- `sphinx.io.SphinxI18nReader.line` is deprecated
- `sphinx.util.i18n.find_catalog_source_file()` has changed; the *gettext_compact* argument has been deprecated
- #5403²⁵³³: `sphinx.util.images.guess_mimetype()` has changed; the *content* argument has been deprecated

1.8.0b1

- *source_parsers* is deprecated
- *autodoc_default_flags* is deprecated
- quickstart: `--epub` option becomes default, so it is deprecated
- Drop function based directive support. For now, Sphinx only supports class based directives (see *Directive*)
- `sphinx.util.docutils.directive_helper()` is deprecated
- `sphinx.cmdline` is deprecated
- `sphinx.make_mode` is deprecated
- `sphinx.locale.l_()` is deprecated
- #2157²⁵³⁴: helper function `warn()` for HTML themes is deprecated
- `app.override_domain()` is deprecated
- `app.add_stylesheet()` is deprecated
- `app.add_javascript()` is deprecated
- `app.import_object()` is deprecated
- `app.add_source_parser()` has changed; the *suffix* argument has been deprecated
- `sphinx.versioning.prepare()` is deprecated
- `Config.__init__()` has changed; the *dirname*, *filename* and *tags* argument has been deprecated
- `Config.check_types()` is deprecated
- `Config.check_unicode()` is deprecated
- `sphinx.application.CONFIG_FILENAME` is deprecated
- `highlightlang` directive is deprecated
- `BuildEnvironment.load()` is deprecated

²⁵³³ <https://github.com/sphinx-doc/sphinx/issues/5403>

²⁵³⁴ <https://github.com/sphinx-doc/sphinx/issues/2157>

- `BuildEnvironment.loads()` is deprecated
- `BuildEnvironment.frompickle()` is deprecated
- `env.read_doc()` is deprecated
- `env.update()` is deprecated
- `env._read_serial()` is deprecated
- `env._read_parallel()` is deprecated
- `env.write_doctree()` is deprecated
- `env._nitpick_ignore` is deprecated
- `env.versionchanges` is deprecated
- `env.dump()` is deprecated
- `env.dumps()` is deprecated
- `env.topickle()` is deprecated
- `env.note_versionchange()` is deprecated
- `sphinx.writers.latex.Table.caption_footnotetexts` is deprecated
- `sphinx.writers.latex.Table.header_footnotetexts` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.footnotestack` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.in_container_literal_block` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.next_section_ids` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.next_hyperlink_ids` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.restrict_footnote()` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.unrestrict_footnote()` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.push_hyperlink_ids()` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.pop_hyperlink_ids()` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.check_latex_elements()` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.bibitems` is deprecated
- `sphinx.writers.latex.LaTeXTranslator.hlsettingstack` is deprecated
- `sphinx.writers.latex.ExtBabel.get_shorthandoff()` is deprecated
- `sphinx.writers.html.HTMLTranslator.highlightlang` is deprecated
- `sphinx.writers.html.HTMLTranslator.highlightlang_base` is deprecated
- `sphinx.writers.html.HTMLTranslator.highlightlangopts` is deprecated

- `sphinx.writers.html.HTMLTranslator.highlightlinethreshold` is deprecated
- `sphinx.writers.html5.HTMLTranslator.highlightlang` is deprecated
- `sphinx.writers.html5.HTMLTranslator.highlightlang_base` is deprecated
- `sphinx.writers.html5.HTMLTranslator.highlightlangopts` is deprecated
- `sphinx.writers.html5.HTMLTranslator.highlightlinethreshold` is deprecated
- `sphinx.ext.mathbase` extension is deprecated
- `sphinx.ext.mathbase.math` node is deprecated
- `sphinx.ext.mathbase.displaymath` node is deprecated
- `sphinx.ext.mathbase.eqref` node is deprecated
- `sphinx.ext.mathbase.is_in_section_title()` is deprecated
- `sphinx.ext.mathbase.MathDomain` is deprecated
- `sphinx.ext.mathbase.MathDirective` is deprecated
- `sphinx.ext.mathbase.math_role` is deprecated
- `sphinx.ext.mathbase.setup_math()` is deprecated
- `sphinx.directives.other.VersionChanges` is deprecated
- `sphinx.highlighting.PygmentsBridge.unhighlight()` is deprecated
- `sphinx.ext.mathbase.get_node_equation_number()` is deprecated
- `sphinx.ext.mathbase.wrap_displaymath()` is deprecated
- The `trim_doctest_flags` argument of `sphinx.highlighting.PygmentsBridge` is deprecated

For more details, see [deprecation APIs list](#).

Funzionalità aggiunte

1.8.0b2

- [#5388](#)²⁵³⁵: Ensure frozen object descriptions are reproducible
- [#5362](#)²⁵³⁶: apidoc: Add `--tocfile` option to change the filename of ToC

1.8.0b1

- Add `config-inited` event
- Add `sphinx.config.Any` to represent the config value accepts any type of value

²⁵³⁵ <https://github.com/sphinx-doc/sphinx/issues/5388>

²⁵³⁶ <https://github.com/sphinx-doc/sphinx/issues/5362>

- `source_suffix` allows a mapping fileext to file types
- Add `author` as a configuration value
- #2852²⁵³⁷: `imgconverter`: Support to convert GIF to PNG
- `sphinx-build` command supports `il8n` console output
- Add `app.add_message_catalog()` and `sphinx.locale.get_translations()` to support translation for 3rd party extensions
- helper function `warning()` for HTML themes is added
- Add `Domain.enumerable_nodes` to manage own enumerable nodes for domains (experimental)
- Add a new keyword argument `override` to Application APIs
- LaTeX: new key `'fvset'` for `latex_elements`. For XeLaTeX/LuaLaTeX its default sets `fanvyrb` to use normal, not small, fontsize in code-blocks (refs: #4793²⁵³⁸)
- Add `html_css_files` and `epub_css_files` for adding CSS files from configuration
- Add `html_js_files` for adding JS files from configuration
- #4834²⁵³⁹: Ensure set object descriptions are reproducible.
- #4828²⁵⁴⁰: Allow to override `numfig_format` partially. Full definition is not needed.
- Improve warning messages during including (refs: #4818²⁵⁴¹)
- LaTeX: separate customizability of `guilabel` and `menuselection` (refs: #4830²⁵⁴²)
- Add `Config.read()` classmethod to create a new config object from configuration file
- #4866²⁵⁴³: Wrap graphviz diagrams in `<div>` tag
- `viewcode`: Add `viewcode-find-source` and `viewcode-follow-imported` to load source code without loading
- #4785²⁵⁴⁴: `napoleon`: Add strings to translation file for localisation
- #4927²⁵⁴⁵: Display a warning when invalid values are passed to `linenothreshold` option of `highlight` directive
- C++:
 - Add a `cpp:texpr` role as a sibling to `cpp:expr`.
 - Add support for unions.

²⁵³⁷ <https://github.com/sphinx-doc/sphinx/issues/2852>

²⁵³⁸ <https://github.com/sphinx-doc/sphinx/issues/4793>

²⁵³⁹ <https://github.com/sphinx-doc/sphinx/issues/4834>

²⁵⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/4828>

²⁵⁴¹ <https://github.com/sphinx-doc/sphinx/issues/4818>

²⁵⁴² <https://github.com/sphinx-doc/sphinx/issues/4830>

²⁵⁴³ <https://github.com/sphinx-doc/sphinx/issues/4866>

²⁵⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/4785>

²⁵⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/4927>

- #3593²⁵⁴⁶, #2683²⁵⁴⁷: add support for anonymous entities using names starting with @.
- #5147²⁵⁴⁸: add support for (most) character literals.
- Cross-referencing entities inside primary templates is supported, and now properly documented.
- #1552²⁵⁴⁹: add new cross-referencing format for `cpp:any` and `cpp:func` roles, for referencing specific function overloads.
- #3606²⁵⁵⁰: MathJax should be loaded with `async` attribute
- `html`: Output `canonical_url` metadata if `html_baseurl` set (refs: #4193²⁵⁵¹)
- #5029²⁵⁵²: `autosummary`: expose `inherited_members` to template
- #3784²⁵⁵³: `mathjax`: Add `mathjax_options` to give options to script tag for mathjax
- #726²⁵⁵⁴, #969²⁵⁵⁵: `mathjax`: Add `mathjax_config` to give in-line configurations for mathjax
- #4362²⁵⁵⁶: `latex`: Don't overwrite `.tex` file if document not changed
- #1431²⁵⁵⁷: `latex`: Add alphanumeric enumerated list support
- Add `latex_use_xindy` for UTF-8 savvy indexing, defaults to `True` if `latex_engine` is `'xelatex'` or `'lualatex'`. (refs: #5134²⁵⁵⁸, #5192²⁵⁵⁹, #5212²⁵⁶⁰)
- #4976²⁵⁶¹: `SphinxLoggerAdapter.info()` now supports `location` parameter
- #5122²⁵⁶²: `setuptools`: support `nitpicky` option
- #2820²⁵⁶³: `autoclass` directive supports nested class
- Add `app.add_html_math_renderer()` to register a math renderer for HTML
- Apply `trim_doctest_flags` to all builders (cf. `text`, `manpages`)
- #5140²⁵⁶⁴: `linkcheck`: Add better `Accept` header to HTTP client

²⁵⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/3593>

²⁵⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/2683>

²⁵⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/5147>

²⁵⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/1552>

²⁵⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/3606>

²⁵⁵¹ <https://github.com/sphinx-doc/sphinx/issues/4193>

²⁵⁵² <https://github.com/sphinx-doc/sphinx/issues/5029>

²⁵⁵³ <https://github.com/sphinx-doc/sphinx/issues/3784>

²⁵⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/726>

²⁵⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/969>

²⁵⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/4362>

²⁵⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/1431>

²⁵⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/5134>

²⁵⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/5192>

²⁵⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/5212>

²⁵⁶¹ <https://github.com/sphinx-doc/sphinx/issues/4976>

²⁵⁶² <https://github.com/sphinx-doc/sphinx/issues/5122>

²⁵⁶³ <https://github.com/sphinx-doc/sphinx/issues/2820>

²⁵⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/5140>

- [#4614](#)²⁵⁶⁵: sphinx-build: Add `--keep-going` option to show all warnings
- Add `math:numref` role to refer equations (Same as `eq`)
- quickstart: epub builder is enabled by default
- [#5246](#)²⁵⁶⁶: Add `singlehtml_sidebars` to configure sidebars for singlehtml builder
- [#5273](#)²⁵⁶⁷: doctest: Skip doctest conditionally
- [#5306](#)²⁵⁶⁸: autodoc: emit a warning for invalid typehints
- [#4075](#)²⁵⁶⁹, [#5215](#)²⁵⁷⁰: autodoc: Add `autodoc_default_options` which accepts option values as dict

Bugs fixed

1.8.0b2

- html: search box overrides to other elements if scrolled
- i18n: warnings for translation catalogs have wrong line numbers (refs: [#5321](#)²⁵⁷¹)
- [#5325](#)²⁵⁷²: latex: cross references has been broken by multiply labeled objects
- C++, fixes for symbol addition and lookup. Lookup should no longer break in partial builds. See also [#5337](#)²⁵⁷³.
- [#5348](#)²⁵⁷⁴: download reference to remote file is not displayed
- [#5282](#)²⁵⁷⁵: html theme: `pygments_style` of theme was overridden by `conf.py` by default
- [#4379](#)²⁵⁷⁶: toctree shows confusing warning when document is excluded
- [#2401](#)²⁵⁷⁷: autodoc: `:members:` causes `:special-members:` not to be shown
- autodoc: ImportError is replaced by AttributeError for deeper module
- [#2720](#)²⁵⁷⁸, [#4034](#)²⁵⁷⁹: Incorrect links with `:download:`, duplicate names, and parallel builds

²⁵⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/4614>

²⁵⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/5246>

²⁵⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/5273>

²⁵⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/5306>

²⁵⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/4075>

²⁵⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/5215>

²⁵⁷¹ <https://github.com/sphinx-doc/sphinx/issues/5321>

²⁵⁷² <https://github.com/sphinx-doc/sphinx/issues/5325>

²⁵⁷³ <https://github.com/sphinx-doc/sphinx/issues/5337>

²⁵⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/5348>

²⁵⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/5282>

²⁵⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/4379>

²⁵⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/2401>

²⁵⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/2720>

²⁵⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/4034>

- [#5290²⁵⁸⁰](#): autodoc: failed to analyze source code in egg package
- [#5399²⁵⁸¹](#): Sphinx crashes if unknown po file exists

1.8.0b1

- i18n: message catalogs were reset on each initialization
- [#4850²⁵⁸²](#): latex: footnote inside footnote was not rendered
- [#4945²⁵⁸³](#): i18n: fix lang_COUNTRY not fallback correctly for IndexBuilder. Thanks to Shengjing Zhu.
- [#4983²⁵⁸⁴](#): productionlist directive generates invalid IDs for the tokens
- [#5132²⁵⁸⁵](#): lualatex: PDF build fails if indexed word starts with Unicode character
- [#5133²⁵⁸⁶](#): latex: index headings "Symbols" and "Numbers" not internationalized
- [#5114²⁵⁸⁷](#): sphinx-build: Handle errors on scanning documents
- epub: spine has been broken when "self" is listed on toctree (refs: [#4611²⁵⁸⁸](#))
- [#344²⁵⁸⁹](#): autosummary does not understand docstring of module level attributes
- [#5191²⁵⁹⁰](#): C++, prevent nested declarations in functions to avoid lookup problems.
- [#5126²⁵⁹¹](#): C++, add missing isPack method for certain template parameter types.
- [#5187²⁵⁹²](#): C++, parse attributes on declarators as well.
- C++, parse delete expressions and basic new expressions as well.
- [#5002²⁵⁹³](#): graphviz: SVGs do not adapt to the column width

²⁵⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/5290>

²⁵⁸¹ <https://github.com/sphinx-doc/sphinx/issues/5399>

²⁵⁸² <https://github.com/sphinx-doc/sphinx/issues/4850>

²⁵⁸³ <https://github.com/sphinx-doc/sphinx/issues/4945>

²⁵⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/4983>

²⁵⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/5132>

²⁵⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/5133>

²⁵⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/5114>

²⁵⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/4611>

²⁵⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/344>

²⁵⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/5191>

²⁵⁹¹ <https://github.com/sphinx-doc/sphinx/issues/5126>

²⁵⁹² <https://github.com/sphinx-doc/sphinx/issues/5187>

²⁵⁹³ <https://github.com/sphinx-doc/sphinx/issues/5002>

Features removed

1.8.0b1

- `sphinx.ext.pngmath` extension

Documentation

1.8.0b1

- [#5083](#)²⁵⁹⁴: Fix wrong `make.bat` option for internationalization.
- [#5115](#)²⁵⁹⁵: `napoleon`: add admonitions added by [#4613](#)²⁵⁹⁶ to the docs.

14.99 Release 1.7.9 (released Sep 05, 2018)

Funzionalità aggiunte

- [#5359](#)²⁵⁹⁷: Make generated texinfo files reproducible by sorting the anchors

Bugs fixed

- [#5361](#)²⁵⁹⁸: crashed on incremental build if document uses `include` directive

14.100 Release 1.7.8 (released Aug 29, 2018)

Modifiche incompatibili

- The type of `env.included` has been changed to dict of set

²⁵⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/5083>

²⁵⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/5115>

²⁵⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/4613>

²⁵⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/5359>

²⁵⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/5361>

Bugs fixed

- [#5320](#)²⁵⁹⁹: intersphinx: crashed if invalid url given
- [#5326](#)²⁶⁰⁰: manpage: crashed when invalid docname is specified as `man_pages`
- [#5322](#)²⁶⁰¹: autodoc: Any typehint causes formatting error
- [#5327](#)²⁶⁰²: "document isn't included in any toctree" warning on rebuild with generated files
- [#5335](#)²⁶⁰³: quickstart: escape sequence has been displayed with MacPorts' python

14.101 Release 1.7.7 (released Aug 19, 2018)

Bugs fixed

- [#5198](#)²⁶⁰⁴: document not in toctree warning when including files only for parallel builds
- LaTeX: reduce "Token not allowed in a PDF string" hyperref warnings in latex console output (refs: [#5236](#)²⁶⁰⁵)
- LaTeX: suppress "remreset Warning: The remreset package is obsolete" in latex console output with recent LaTeX (refs: [#5237](#)²⁶⁰⁶)
- [#5234](#)²⁶⁰⁷: PDF output: usage of PAPER environment variable is broken since Sphinx 1.5
- LaTeX: fix the `latex_engine` documentation regarding Latin Modern font with XeLaTeX/LuaLaTeX (refs: [#5251](#)²⁶⁰⁸)
- [#5280](#)²⁶⁰⁹: autodoc: Fix wrong type annotations for complex typing
- autodoc: Optional types are wrongly rendered
- [#5291](#)²⁶¹⁰: autodoc crashed by ForwardRef types
- [#5211](#)²⁶¹¹: autodoc: No docs generated for `functools.partial` functions
- [#5306](#)²⁶¹²: autodoc: `getargspec()` raises `NameError` for invalid typehints

²⁵⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/5320>

²⁶⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/5326>

²⁶⁰¹ <https://github.com/sphinx-doc/sphinx/issues/5322>

²⁶⁰² <https://github.com/sphinx-doc/sphinx/issues/5327>

²⁶⁰³ <https://github.com/sphinx-doc/sphinx/issues/5335>

²⁶⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/5198>

²⁶⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/5236>

²⁶⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/5237>

²⁶⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/5234>

²⁶⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/5251>

²⁶⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/5280>

²⁶¹⁰ <https://github.com/sphinx-doc/sphinx/issues/5291>

²⁶¹¹ <https://github.com/sphinx-doc/sphinx/issues/5211>

²⁶¹² <https://github.com/sphinx-doc/sphinx/issues/5306>

- #5298²⁶¹³: imgmath: math_number_all causes equations to have two numbers in html
- #5294²⁶¹⁴: sphinx-quickstart blank prompts in PowerShell

14.102 Release 1.7.6 (released Jul 17, 2018)

Bugs fixed

- #5037²⁶¹⁵: LaTeX \sphinxupquote{} breaks in Russian
- sphinx.testing uses deprecated pytest API; Node.get_marker(name)
- #5016²⁶¹⁶: crashed when recomcommonmark.AutoStrictify is enabled
- #5022²⁶¹⁷: latex: crashed with Docutils package provided by Debian/Ubuntu
- #5009²⁶¹⁸: latex: a label for table is vanished if table does not have a caption
- #5048²⁶¹⁹: crashed with numbered toctree
- #2410²⁶²⁰: C, render empty argument lists for macros.
- C++, fix lookup of full template specializations with no template arguments.
- #4667²⁶²¹: C++, fix assertion on missing references in global scope when using intersphinx. Thanks to Alan M. Carroll.
- #5019²⁶²²: autodoc: crashed by Form Feed Character
- #5032²⁶²³: autodoc: loses the first staticmethod parameter for old styled classes
- #5036²⁶²⁴: quickstart: Typing Ctrl-U clears the whole of line
- #5066²⁶²⁵: html: "relations" sidebar is not shown by default
- #5091²⁶²⁶: latex: curly braces in index entries are not handled correctly
- #5070²⁶²⁷: epub: Wrong internal href fragment links
- #5104²⁶²⁸: apidoc: Interface of sphinx.apidoc:main() has changed

²⁶¹³ <https://github.com/sphinx-doc/sphinx/issues/5298>

²⁶¹⁴ <https://github.com/sphinx-doc/sphinx/issues/5294>

²⁶¹⁵ <https://github.com/sphinx-doc/sphinx/issues/5037>

²⁶¹⁶ <https://github.com/sphinx-doc/sphinx/issues/5016>

²⁶¹⁷ <https://github.com/sphinx-doc/sphinx/issues/5022>

²⁶¹⁸ <https://github.com/sphinx-doc/sphinx/issues/5009>

²⁶¹⁹ <https://github.com/sphinx-doc/sphinx/issues/5048>

²⁶²⁰ <https://github.com/sphinx-doc/sphinx/issues/2410>

²⁶²¹ <https://github.com/sphinx-doc/sphinx/issues/4667>

²⁶²² <https://github.com/sphinx-doc/sphinx/issues/5019>

²⁶²³ <https://github.com/sphinx-doc/sphinx/issues/5032>

²⁶²⁴ <https://github.com/sphinx-doc/sphinx/issues/5036>

²⁶²⁵ <https://github.com/sphinx-doc/sphinx/issues/5066>

²⁶²⁶ <https://github.com/sphinx-doc/sphinx/issues/5091>

²⁶²⁷ <https://github.com/sphinx-doc/sphinx/issues/5070>

²⁶²⁸ <https://github.com/sphinx-doc/sphinx/issues/5104>

- [#4272](#)²⁶²⁹: PDF builds of French projects have issues with XeTeX
- [#5076](#)²⁶³⁰: napoleon raises RuntimeError with python 3.7
- [#5125](#)²⁶³¹: sphinx-build: Interface of `sphinx:main()` has changed
- sphinx-build: `sphinx.cmd.build.main()` refers `sys.argv` instead of given argument
- [#5146](#)²⁶³²: autosummary: warning is emitted when the first line of docstring ends with literal notation
- autosummary: warnings of autosummary indicates wrong location (refs: [#5146](#)²⁶³³)
- [#5143](#)²⁶³⁴: autodoc: crashed on inspecting dict like object which does not support sorting
- [#5139](#)²⁶³⁵: autodoc: Enum argument missing if it shares value with another
- [#4946](#)²⁶³⁶: py domain: rtype field could not handle "None" as a type
- [#5176](#)²⁶³⁷: LaTeX: indexing of terms containing @, !, or " fails
- [#5161](#)²⁶³⁸: html: crashes if copying static files are failed
- [#5167](#)²⁶³⁹: autodoc: Fix formatting type annotations for tuples with more than two arguments
- [#3329](#)²⁶⁴⁰: i18n: crashed by auto-symbol footnote references
- [#5158](#)²⁶⁴¹: autosummary: module summary has been broken when it starts with heading

14.103 Release 1.7.5 (released May 29, 2018)

Bugs fixed

- [#4924](#)²⁶⁴²: html search: Upper characters problem in any other languages
- [#4932](#)²⁶⁴³: apidoc: some subpackage is ignored if sibling subpackage contains a module starting with underscore

²⁶²⁹ <https://github.com/sphinx-doc/sphinx/issues/4272>

²⁶³⁰ <https://github.com/sphinx-doc/sphinx/issues/5076>

²⁶³¹ <https://github.com/sphinx-doc/sphinx/issues/5125>

²⁶³² <https://github.com/sphinx-doc/sphinx/issues/5146>

²⁶³³ <https://github.com/sphinx-doc/sphinx/issues/5146>

²⁶³⁴ <https://github.com/sphinx-doc/sphinx/issues/5143>

²⁶³⁵ <https://github.com/sphinx-doc/sphinx/issues/5139>

²⁶³⁶ <https://github.com/sphinx-doc/sphinx/issues/4946>

²⁶³⁷ <https://github.com/sphinx-doc/sphinx/issues/5176>

²⁶³⁸ <https://github.com/sphinx-doc/sphinx/issues/5161>

²⁶³⁹ <https://github.com/sphinx-doc/sphinx/issues/5167>

²⁶⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/3329>

²⁶⁴¹ <https://github.com/sphinx-doc/sphinx/issues/5158>

²⁶⁴² <https://github.com/sphinx-doc/sphinx/issues/4924>

²⁶⁴³ <https://github.com/sphinx-doc/sphinx/issues/4932>

- [#4863](#)²⁶⁴⁴, [#4938](#)²⁶⁴⁵, [#4939](#)²⁶⁴⁶: i18n doesn't handle correctly node.title as used for contents, topic, admonition, table and section.
- [#4913](#)²⁶⁴⁷: i18n: literal blocks in bullet list are not translated
- [#4962](#)²⁶⁴⁸: C++, raised TypeError on duplicate declaration.
- [#4825](#)²⁶⁴⁹: C++, properly parse expr roles and give better error messages when (escaped) line breaks are present.
- C++, properly use desc_addname nodes for prefixes of names.
- C++, parse pack expansions in function calls.
- [#4915](#)²⁶⁵⁰, [#4916](#)²⁶⁵¹: links on search page are broken when using dirhtml builder
- [#4969](#)²⁶⁵²: autodoc: constructor method should not have return annotation
- latex: deeply nested enumerated list which is beginning with non-1 causes LaTeX engine crashed
- [#4978](#)²⁶⁵³: latex: shorthandoff is not set up for Brazil locale
- [#4928](#)²⁶⁵⁴: i18n: Ignore dot-directories like .git/ in LC_MESSAGES/
- [#4946](#)²⁶⁵⁵: py domain: type field could not handle "None" as a type
- [#4979](#)²⁶⁵⁶: latex: Incorrect escaping of curly braces in index entries
- [#4956](#)²⁶⁵⁷: autodoc: Failed to extract document from a subclass of the class on mocked module
- [#4973](#)²⁶⁵⁸: latex: glossary directive adds whitespace to each item
- [#4980](#)²⁶⁵⁹: latex: Explicit labels on code blocks are duplicated
- [#4919](#)²⁶⁶⁰: node.asdom() crashes if toctree has :numbered: option
- [#4914](#)²⁶⁶¹: autodoc: Parsing error when using dataclasses without default values
- [#4931](#)²⁶⁶²: autodoc: crashed when handler for autodoc-skip-member raises an error

²⁶⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/4863>

²⁶⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/4938>

²⁶⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/4939>

²⁶⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/4913>

²⁶⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/4962>

²⁶⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/4825>

²⁶⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/4915>

²⁶⁵¹ <https://github.com/sphinx-doc/sphinx/issues/4916>

²⁶⁵² <https://github.com/sphinx-doc/sphinx/issues/4969>

²⁶⁵³ <https://github.com/sphinx-doc/sphinx/issues/4978>

²⁶⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/4928>

²⁶⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/4946>

²⁶⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/4979>

²⁶⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/4956>

²⁶⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/4973>

²⁶⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/4980>

²⁶⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/4919>

²⁶⁶¹ <https://github.com/sphinx-doc/sphinx/issues/4914>

²⁶⁶² <https://github.com/sphinx-doc/sphinx/issues/4931>

- [#4931²⁶⁶³](#): autodoc: crashed when subclass of mocked class are processed by napoleon module
- [#5007²⁶⁶⁴](#): sphinx-build crashes when error log contains a "%" character

14.104 Release 1.7.4 (released Apr 25, 2018)

Bugs fixed

- [#4885²⁶⁶⁵](#), [#4887²⁶⁶⁶](#): domains: Crashed with duplicated objects
- [#4889²⁶⁶⁷](#): latex: sphinx.writers.latex causes recursive import

14.105 Release 1.7.3 (released Apr 23, 2018)

Bugs fixed

- [#4769²⁶⁶⁸](#): autodoc loses the first staticmethod parameter
- [#4790²⁶⁶⁹](#): autosummary: too wide two column tables in PDF builds
- [#4795²⁶⁷⁰](#): Latex customization via `_templates/longtable.tex_t` is broken
- [#4789²⁶⁷¹](#): imgconverter: confused by convert.exe of Windows
- [#4783²⁶⁷²](#): On windows, Sphinx crashed when drives of srcdir and outdir are different
- [#4812²⁶⁷³](#): autodoc ignores type annotated variables
- [#4817²⁶⁷⁴](#): wrong URLs on warning messages
- [#4784²⁶⁷⁵](#): latex: `latex_show_urls` assigns incorrect footnote numbers if hyperlinks exists inside substitutions
- [#4837²⁶⁷⁶](#): latex with class memoir Error: Font command `\sf` is not supported

²⁶⁶³ <https://github.com/sphinx-doc/sphinx/issues/4931>

²⁶⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/5007>

²⁶⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/4885>

²⁶⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/4887>

²⁶⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/4889>

²⁶⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/4769>

²⁶⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/4790>

²⁶⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/4795>

²⁶⁷¹ <https://github.com/sphinx-doc/sphinx/issues/4789>

²⁶⁷² <https://github.com/sphinx-doc/sphinx/issues/4783>

²⁶⁷³ <https://github.com/sphinx-doc/sphinx/issues/4812>

²⁶⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/4817>

²⁶⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/4784>

²⁶⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/4837>

- [#4803](#)²⁶⁷⁷: latex: too slow in proportion to number of auto numbered footnotes
- [#4838](#)²⁶⁷⁸: htmlhelp: The entries in .hhp file is not ordered
- toctree directive tries to glob for URL having query_string
- [#4871](#)²⁶⁷⁹: html search: Upper characters problem in German
- [#4717](#)²⁶⁸⁰: latex: Compilation for German docs failed with LuaLaTeX and XeLaTeX
- [#4459](#)²⁶⁸¹: duplicated labels detector does not work well in parallel build
- [#4878](#)²⁶⁸²: Crashed with extension which returns invalid metadata

14.106 Release 1.7.2 (released Mar 21, 2018)

Modifiche incompatibili

- [#4520](#)²⁶⁸³: apidoc: folders with an empty __init__.py are no longer excluded from TOC

Bugs fixed

- [#4669](#)²⁶⁸⁴: sphinx.build_main and sphinx.make_main throw NameError
- [#4685](#)²⁶⁸⁵: autosummary emits meaningless warnings
- autodoc: crashed when invalid options given
- pydomain: always strip parenthesis if empty (refs: [#1042](#)²⁶⁸⁶)
- [#4689](#)²⁶⁸⁷: autosummary: unexpectedly strips docstrings containing "i.e."
- [#4701](#)²⁶⁸⁸: viewcode: Misplaced <div> in viewcode html output
- [#4444](#)²⁶⁸⁹: Don't require numfig to use :numref: on sections
- [#4727](#)²⁶⁹⁰: Option clash for package textcomp
- [#4725](#)²⁶⁹¹: Sphinx does not work with python 3.5.0 and 3.5.1

²⁶⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/4803>

²⁶⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/4838>

²⁶⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/4871>

²⁶⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/4717>

²⁶⁸¹ <https://github.com/sphinx-doc/sphinx/issues/4459>

²⁶⁸² <https://github.com/sphinx-doc/sphinx/issues/4878>

²⁶⁸³ <https://github.com/sphinx-doc/sphinx/issues/4520>

²⁶⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/4669>

²⁶⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/4685>

²⁶⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/1042>

²⁶⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/4689>

²⁶⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/4701>

²⁶⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/4444>

²⁶⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/4727>

²⁶⁹¹ <https://github.com/sphinx-doc/sphinx/issues/4725>

- [#4716²⁶⁹²](#): Generation PDF file with TexLive on Windows, file not found error
- [#4574²⁶⁹³](#): vertical space before equation in latex
- [#4720²⁶⁹⁴](#): message when an image is mismatched for builder is not clear
- [#4655²⁶⁹⁵](#), [#4684²⁶⁹⁶](#): Incomplete localization strings in Polish and Chinese
- [#2286²⁶⁹⁷](#): Sphinx crashes when error is happens in rendering HTML pages
- [#4688²⁶⁹⁸](#): Error to download remote images having long URL
- [#4754²⁶⁹⁹](#): sphinx/pycode/__init__.py raises AttributeError
- [#1435²⁷⁰⁰](#): qthelp builder should htmlescape keywords
- epub: Fix docTitle elements of toc.ncx is not escaped
- [#4520²⁷⁰¹](#): apidoc: Subpackage not in toc (introduced in 1.6.6) now fixed
- [#4767²⁷⁰²](#): html: search highlighting breaks mathjax equations

14.107 Release 1.7.1 (released Feb 23, 2018)

Deprecato

- [#4623²⁷⁰³](#): `sphinx.build_main()` is deprecated.
- autosummary: The interface of `sphinx.ext.autosummary.get_documenter()` has been changed (Since 1.7.0)
- [#4664²⁷⁰⁴](#): `sphinx.ext.intersphinx.debug()` is deprecated.

For more details, see [deprecation APIs list](#).

²⁶⁹² <https://github.com/sphinx-doc/sphinx/issues/4716>

²⁶⁹³ <https://github.com/sphinx-doc/sphinx/issues/4574>

²⁶⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/4720>

²⁶⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/4655>

²⁶⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/4684>

²⁶⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/2286>

²⁶⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/4688>

²⁶⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/4754>

²⁷⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/1435>

²⁷⁰¹ <https://github.com/sphinx-doc/sphinx/issues/4520>

²⁷⁰² <https://github.com/sphinx-doc/sphinx/issues/4767>

²⁷⁰³ <https://github.com/sphinx-doc/sphinx/issues/4623>

²⁷⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/4664>

Bugs fixed

- #4608²⁷⁰⁵: epub: Invalid meta tag is generated
- #4260²⁷⁰⁶: autodoc: keyword only argument separator is not disappeared if it is appeared at top of the argument list
- #4622²⁷⁰⁷: epub: `epub_scheme` does not effect to content.opf
- #4627²⁷⁰⁸: graphviz: Fit graphviz images to page
- #4617²⁷⁰⁹: quickstart: PROJECT_DIR argument is required
- #4623²⁷¹⁰: sphinx.build_main no longer exists in 1.7.0
- #4615²⁷¹¹: The argument of `sphinx.build` has been changed in 1.7.0
- autosummary: The interface of `sphinx.ext.autosummary.get_documenter()` has been changed
- #4630²⁷¹²: Have order on msgids in sphinx.pot deterministic
- #4563²⁷¹³: autosummary: Incorrect end of line punctuation detection
- #4577²⁷¹⁴: Enumerated sublists with explicit start with wrong number
- #4641²⁷¹⁵: A external link in TOC cannot contain "?" with `:glob:` option
- C++, add missing parsing of explicit casts and typeid in expression parsing.
- C++, add missing parsing of `this` in expression parsing.
- #4655²⁷¹⁶: Fix incomplete localization strings in Polish
- #4653²⁷¹⁷: Fix error reporting for parameterless ImportErrors
- #4664²⁷¹⁸: Reading objects.inv fails again
- #4662²⁷¹⁹: any refs with `term` targets crash when an ambiguity is encountered

²⁷⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/4608>

²⁷⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/4260>

²⁷⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/4622>

²⁷⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/4627>

²⁷⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/4617>

²⁷¹⁰ <https://github.com/sphinx-doc/sphinx/issues/4623>

²⁷¹¹ <https://github.com/sphinx-doc/sphinx/issues/4615>

²⁷¹² <https://github.com/sphinx-doc/sphinx/issues/4630>

²⁷¹³ <https://github.com/sphinx-doc/sphinx/issues/4563>

²⁷¹⁴ <https://github.com/sphinx-doc/sphinx/issues/4577>

²⁷¹⁵ <https://github.com/sphinx-doc/sphinx/issues/4641>

²⁷¹⁶ <https://github.com/sphinx-doc/sphinx/issues/4655>

²⁷¹⁷ <https://github.com/sphinx-doc/sphinx/issues/4653>

²⁷¹⁸ <https://github.com/sphinx-doc/sphinx/issues/4664>

²⁷¹⁹ <https://github.com/sphinx-doc/sphinx/issues/4662>

14.108 Release 1.7.0 (released Feb 12, 2018)

Dipendenze

1.7.0b1

- Add packaging package

Modifiche incompatibili

1.7.0b1

- [#3668](#)²⁷²⁰: The arguments has changed of main functions for each command
- [#3893](#)²⁷²¹: Unknown `html_theme_options` throw warnings instead of errors
- [#3927](#)²⁷²²: Python parameter/variable types should match classes, not all objects
- [#3962](#)²⁷²³: sphinx-apidoc now recognizes given directory as an implicit namespace package when `--implicit-namespaces` option given, not subdirectories of given directory.
- [#3929](#)²⁷²⁴: apidoc: Move `sphinx.apidoc` to `sphinx.ext.apidoc`
- [#4226](#)²⁷²⁵: apidoc: Generate new style makefile (make-mode)
- [#4274](#)²⁷²⁶: sphinx-build returns 2 as an exit code on argument error
- [#4389](#)²⁷²⁷: output directory will be created after loading extensions
- autodoc does not generate warnings messages to the generated document even if `keep_warnings` is `True`. They are only emitted to `stderr`.
- shebang line is removed from generated `conf.py`
- [#2557](#)²⁷²⁸: autodoc: `autodoc_mock_imports` only mocks specified modules with their descendants. It does not mock their ancestors. If you want to mock them, please specify the name of ancestors explicitly.
- [#3620](#)²⁷²⁹: html theme: move `DOCUMENTATION_OPTIONS` to independent JavaScript file (refs: [#4295](#)²⁷³⁰)
- [#4246](#)²⁷³¹: Limit width of text body for all themes. Configurable via theme options

²⁷²⁰ <https://github.com/sphinx-doc/sphinx/issues/3668>

²⁷²¹ <https://github.com/sphinx-doc/sphinx/issues/3893>

²⁷²² <https://github.com/sphinx-doc/sphinx/issues/3927>

²⁷²³ <https://github.com/sphinx-doc/sphinx/issues/3962>

²⁷²⁴ <https://github.com/sphinx-doc/sphinx/issues/3929>

²⁷²⁵ <https://github.com/sphinx-doc/sphinx/issues/4226>

²⁷²⁶ <https://github.com/sphinx-doc/sphinx/issues/4274>

²⁷²⁷ <https://github.com/sphinx-doc/sphinx/issues/4389>

²⁷²⁸ <https://github.com/sphinx-doc/sphinx/issues/2557>

²⁷²⁹ <https://github.com/sphinx-doc/sphinx/issues/3620>

²⁷³⁰ <https://github.com/sphinx-doc/sphinx/issues/4295>

²⁷³¹ <https://github.com/sphinx-doc/sphinx/issues/4246>

`body_min_width` and `body_max_width`.

- [#4771](#)²⁷³²: apidoc: The `exclude_patterns` arguments are ignored if they are placed just after command line options

1.7.0b2

- [#4467](#)²⁷³³: html theme: Rename `csss` block to `css`

Deprecato

1.7.0b1

- using a string value for `html_sidebars` is deprecated and only list values will be accepted at 2.0.
- `format_annotation()` and `formatargspec()` is deprecated. Please use `sphinx.util.inspect.Signature` instead.
- `sphinx.ext.autodoc.AutodocReporter` is replaced by `sphinx.util.docutils.switch_source_input()` and now deprecated. It will be removed in Sphinx 2.0.
- `sphinx.ext.autodoc.add_documenter()` and `AutoDirective._register` is now deprecated. Please use `app.add_autodocumenter()` instead.
- `AutoDirective._special_attrgetters` is now deprecated. Please use `app.add_autodoc_attrgetter()` instead.

Funzionalità aggiunte

1.7.0b1

- C++, handle `decltype(auto)`.
- [#2406](#)²⁷³⁴: C++, add proper parsing of expressions, including linking of identifiers.
- C++, add a `cpp:expr` role for inserting inline C++ expressions or types.
- C++, support explicit member instantiations with shorthand `template` prefix
- C++, make function parameters linkable, like template params.
- [#3638](#)²⁷³⁵: Allow to change a label of reference to equation using `math_eqref_format`
- Now `suppress_warnings` accepts following configurations:
 - `ref.python` (ref: [#3866](#)²⁷³⁶)

²⁷³² <https://github.com/sphinx-doc/sphinx/issues/4771>

²⁷³³ <https://github.com/sphinx-doc/sphinx/issues/4467>

²⁷³⁴ <https://github.com/sphinx-doc/sphinx/issues/2406>

²⁷³⁵ <https://github.com/sphinx-doc/sphinx/issues/3638>

²⁷³⁶ <https://github.com/sphinx-doc/sphinx/issues/3866>

- [#3872²⁷³⁷](#): Add latex key to configure literal blocks caption position in PDF output (refs [#3792²⁷³⁸](#), [#1723²⁷³⁹](#))
- In case of missing docstring try to retrieve doc from base classes (ref: [#3140²⁷⁴⁰](#))
- [#4023²⁷⁴¹](#): Clarify error message when any role has more than one target.
- [#3973²⁷⁴²](#): epub: allow to override build date
- [#3972²⁷⁴³](#): epub: Sort manifest entries by filename
- [#4052²⁷⁴⁴](#): viewcode: Sort before highlighting module code
- [#1448²⁷⁴⁵](#): qthelp: Add new config value; *qthelp_namespace*
- [#4140²⁷⁴⁶](#): html themes: Make body tag inheritable
- [#4168²⁷⁴⁷](#): improve zh search with jieba
- HTML themes can set up default sidebars through `theme.conf`
- [#3160²⁷⁴⁸](#): html: Use `<kdb>` to represent `:kbd:` role
- [#4212²⁷⁴⁹](#): autosummary: catch all exceptions when importing modules
- [#4166²⁷⁵⁰](#): Add *math_numfig* for equation numbering by section (refs: [#3991²⁷⁵¹](#), [#4080²⁷⁵²](#)). Thanks to Oliver Jahn.
- [#4311²⁷⁵³](#): Let LaTeX obey *numfig_secnum_depth* for figures, tables, and code-blocks
- [#947²⁷⁵⁴](#): autodoc now supports ignore-module-all to ignore a module's `__all__`
- [#4332²⁷⁵⁵](#): Let LaTeX obey *math_numfig* for equation numbering
- [#4093²⁷⁵⁶](#): sphinx-build creates empty directories for unknown targets/builders
- Add `top-classes` option for the `sphinx.ext.inheritance_diagram` extension to limit the scope of inheritance graphs.

²⁷³⁷ <https://github.com/sphinx-doc/sphinx/issues/3872>

²⁷³⁸ <https://github.com/sphinx-doc/sphinx/issues/3792>

²⁷³⁹ <https://github.com/sphinx-doc/sphinx/issues/1723>

²⁷⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/3140>

²⁷⁴¹ <https://github.com/sphinx-doc/sphinx/issues/4023>

²⁷⁴² <https://github.com/sphinx-doc/sphinx/issues/3973>

²⁷⁴³ <https://github.com/sphinx-doc/sphinx/issues/3972>

²⁷⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/4052>

²⁷⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/1448>

²⁷⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/4140>

²⁷⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/4168>

²⁷⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/3160>

²⁷⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/4212>

²⁷⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/4166>

²⁷⁵¹ <https://github.com/sphinx-doc/sphinx/issues/3991>

²⁷⁵² <https://github.com/sphinx-doc/sphinx/issues/4080>

²⁷⁵³ <https://github.com/sphinx-doc/sphinx/issues/4311>

²⁷⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/947>

²⁷⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/4332>

²⁷⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/4093>

- [#4183²⁷⁵⁷](#): doctest: `:pyversion:` option also follows PEP-440 specification
- [#4235²⁷⁵⁸](#): html: Add `manpages_url` to make manpage roles to hyperlinks
- [#3570²⁷⁵⁹](#): autodoc: Do not display 'typing.' module for type hints
- [#4354²⁷⁶⁰](#): sphinx-build now emits finish message. Builders can modify it through `Builder.epilog` attribute
- [#4245²⁷⁶¹](#): html themes: Add language to javascript vars list
- [#4079²⁷⁶²](#): html: Add `notranslate` class to each code-blocks, literals and maths to let Google Translate know they are not translatable
- [#4137²⁷⁶³](#): doctest: doctest block is always highlighted as python console (pycon)
- [#4137²⁷⁶⁴](#): doctest: testcode block is always highlighted as python
- [#3998²⁷⁶⁵](#): text: Assign section numbers by default. You can control it using `text_add_secnumbers` and `text_secnumber_suffix`

1.7.0b2

- [#4271²⁷⁶⁶](#): sphinx-build supports an option called `-j auto` to adjust numbers of processes automatically.
- Napoleon: added option to specify custom section tags.

Features removed

1.7.0b1

- Configuration variables
 - `html_use_smartypants`
 - `latex_keep_old_macro_names`
 - `latex_elements['footer']`
- utility methods of `sphinx.application.Sphinx` class
 - `buildname` (property)
 - `_display_chunk()`
 - `old_status_iterator()`

²⁷⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/4183>

²⁷⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/4235>

²⁷⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/3570>

²⁷⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/4354>

²⁷⁶¹ <https://github.com/sphinx-doc/sphinx/issues/4245>

²⁷⁶² <https://github.com/sphinx-doc/sphinx/issues/4079>

²⁷⁶³ <https://github.com/sphinx-doc/sphinx/issues/4137>

²⁷⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/4137>

²⁷⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/3998>

²⁷⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/4271>

- `status_iterator()`
 - `_directive_helper()`
- utility methods of `sphinx.environment.BuildEnvironment` class
 - `currmodule` (property)
 - `currclass` (property)
- epub2 builder
- `prefix` and `colorfunc` parameter for `warn()`
- `sphinx.util.compat` module
- `sphinx.util.nodes.process_only_nodes()`
- LaTeX environment `notice`, use `sphinxadmonition` instead
- LaTeX `\sphinxstylethead`, use `\sphinxstyletheadfamily`
- C++, support of function concepts. Thanks to mickk-on-cpp.
- Not used and previously not documented LaTeX macros `\shortversion` and `\setshortversion`

Bugs fixed

1.7.0b1

- [#3882](https://github.com/sphinx-doc/sphinx/issues/3882)²⁷⁶⁷: Update the order of files for `HTMLHelp` and `QTHelp`
- [#3962](https://github.com/sphinx-doc/sphinx/issues/3962)²⁷⁶⁸: `sphinx-apidoc` does not recognize implicit namespace packages correctly
- [#4094](https://github.com/sphinx-doc/sphinx/issues/4094)²⁷⁶⁹: C++, allow empty template argument lists.
- C++, also hyperlink types in the name of declarations with qualified names.
- C++, do not add index entries for declarations inside concepts.
- C++, support the template disambiguator for dependent names.
- [#4314](https://github.com/sphinx-doc/sphinx/issues/4314)²⁷⁷⁰: For PDF 'howto' documents, numbering of code-blocks differs from the one of figures and tables
- [#4330](https://github.com/sphinx-doc/sphinx/issues/4330)²⁷⁷¹: PDF 'howto' documents have an incoherent default LaTeX `tocdepth` counter setting
- [#4198](https://github.com/sphinx-doc/sphinx/issues/4198)²⁷⁷²: `autosummary` emits multiple 'autodoc-process-docstring' event. Thanks to Joel Nothman.

²⁷⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/3882>

²⁷⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/3962>

²⁷⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/4094>

²⁷⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/4314>

²⁷⁷¹ <https://github.com/sphinx-doc/sphinx/issues/4330>

²⁷⁷² <https://github.com/sphinx-doc/sphinx/issues/4198>

- #4081²⁷⁷³: Warnings and errors colored the same when building
- latex: Do not display 'Release' label if *release* is not set

1.7.0b2

- #4415²⁷⁷⁴: autodoc classifies inherited classmethods as regular methods
- #4415²⁷⁷⁵: autodoc classifies inherited staticmethods as regular methods
- #4472²⁷⁷⁶: DOCUMENTATION_OPTIONS is not defined
- #4491²⁷⁷⁷: autodoc: prefer `_MockImporter` over other importers in `sys.meta_path`
- #4490²⁷⁷⁸: autodoc: type annotation is broken with python 3.7.0a4+
- utils package is no longer installed
- #3952²⁷⁷⁹: apidoc: module header is too escaped
- #4275²⁷⁸⁰: Formats accepted by `sphinx.util.i18n.format_date` are limited
- #4493²⁷⁸¹: recomcommonmark raises `AttributeError` if `AutoStructify` enabled
- #4209²⁷⁸²: intersphinx: In link title, "v" should be optional if target has no version
- #4230²⁷⁸³: slowdown in writing pages with Sphinx 1.6
- #4522²⁷⁸⁴: epub: document is not rebuilt even if config changed

1.7.0b3

- #4019²⁷⁸⁵: `inheritance_diagram` `AttributeError` stopping make process
- #4531²⁷⁸⁶: autosummary: methods are not treated as attributes
- #4538²⁷⁸⁷: autodoc: `sphinx.ext.autodoc.Options` has been moved
- #4539²⁷⁸⁸: autodoc emits warnings for `partialmethods`
- #4223²⁷⁸⁹: doctest: failing tests reported in wrong file, at wrong line

²⁷⁷³ <https://github.com/sphinx-doc/sphinx/issues/4081>

²⁷⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/4415>

²⁷⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/4415>

²⁷⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/4472>

²⁷⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/4491>

²⁷⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/4490>

²⁷⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/3952>

²⁷⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/4275>

²⁷⁸¹ <https://github.com/sphinx-doc/sphinx/issues/4493>

²⁷⁸² <https://github.com/sphinx-doc/sphinx/issues/4209>

²⁷⁸³ <https://github.com/sphinx-doc/sphinx/issues/4230>

²⁷⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/4522>

²⁷⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/4019>

²⁷⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/4531>

²⁷⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/4538>

²⁷⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/4539>

²⁷⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/4223>

- [#4560](#)²⁷⁹⁰: message catalogs are not compiled if specific filenames are given for `sphinx-build` as arguments (refs: [#4560](#)²⁷⁹⁰)
- [#4027](#)²⁷⁹¹: `sphinx.ext.autosectionlabel` now expects labels to be the same as they are in the raw source; no smart quotes, nothing fancy.
- [#4581](#)²⁷⁹²: `apidoc`: Excluded modules still included

Testing

1.7.0b1

- Add support for Docutils 0.14
- Add tests for the `sphinx.ext.inheritance_diagram` extension.

14.109 Release 1.6.7 (released Feb 04, 2018)

Bugs fixed

- [#1922](#)²⁷⁹³: html search: Upper characters problem in French
- [#4412](#)²⁷⁹⁴: Updated jQuery version from 3.1.0 to 3.2.1
- [#4438](#)²⁷⁹⁵: `math`: `math` with labels with whitespace cause html error
- [#2437](#)²⁷⁹⁶: make full reference for classes, aliased with "alias of"
- [#4434](#)²⁷⁹⁷: pure numbers as link targets produce warning
- [#4477](#)²⁷⁹⁸: Build fails after building specific files
- [#4449](#)²⁷⁹⁹: `apidoc`: include "empty" packages that contain modules
- [#3917](#)²⁸⁰⁰: citation labels are transformed to ellipsis
- [#4501](#)²⁸⁰¹: `graphviz`: epub3 validation error caused if graph is not clickable
- [#4514](#)²⁸⁰²: `graphviz`: workaround for wrong map ID which `graphviz` generates

²⁷⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/4560>

²⁷⁹¹ <https://github.com/sphinx-doc/sphinx/issues/4027>

²⁷⁹² <https://github.com/sphinx-doc/sphinx/issues/4581>

²⁷⁹³ <https://github.com/sphinx-doc/sphinx/issues/1922>

²⁷⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/4412>

²⁷⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/4438>

²⁷⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/2437>

²⁷⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/4434>

²⁷⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/4477>

²⁷⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/4449>

²⁸⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/3917>

²⁸⁰¹ <https://github.com/sphinx-doc/sphinx/issues/4501>

²⁸⁰² <https://github.com/sphinx-doc/sphinx/issues/4514>

- [#4525²⁸⁰³](#): autosectionlabel does not support parallel build
- [#3953²⁸⁰⁴](#): Do not raise warning when there is a working intersphinx inventory
- [#4487²⁸⁰⁵](#): math: ValueError is raised on parallel build. Thanks to jschueller.
- [#2372²⁸⁰⁶](#): autosummary: invalid signatures are shown for type annotated functions
- [#3942²⁸⁰⁷](#): html: table is not aligned to center even if `:align: center`

14.110 Release 1.6.6 (released Jan 08, 2018)

Funzionalità aggiunte

- [#4181²⁸⁰⁸](#): autodoc: Sort dictionary keys when possible
- `VerbatimHighlightColor` is a new *LaTeX* `'sphinxsetup'` key (refs: [#4285²⁸⁰⁹](#))
- Easier customizability of LaTeX macros involved in rendering of code-blocks
- Show traceback if `conf.py` raises an exception (refs: [#4369²⁸¹⁰](#))
- Add `smartquotes` to disable smart quotes through `conf.py` (refs: [#3967²⁸¹¹](#))
- Add `smartquotes_action` and `smartquotes_excludes` (refs: [#4142²⁸¹²](#), [#4357²⁸¹³](#))

Bugs fixed

- [#4334²⁸¹⁴](#): sphinx-apidoc: Don't generate references to non-existing files in TOC
- [#4206²⁸¹⁵](#): latex: reST label between paragraphs loses paragraph break
- [#4231²⁸¹⁶](#): html: Apply `fixFirefoxAnchorBug` only under Firefox
- [#4221²⁸¹⁷](#): napoleon depends on autodoc, but users need to load it manually
- [#2298²⁸¹⁸](#): automodule fails to document a class attribute

²⁸⁰³ <https://github.com/sphinx-doc/sphinx/issues/4525>

²⁸⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/3953>

²⁸⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/4487>

²⁸⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/2372>

²⁸⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/3942>

²⁸⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/4181>

²⁸⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/4285>

²⁸¹⁰ <https://github.com/sphinx-doc/sphinx/issues/4369>

²⁸¹¹ <https://github.com/sphinx-doc/sphinx/issues/3967>

²⁸¹² <https://github.com/sphinx-doc/sphinx/issues/4142>

²⁸¹³ <https://github.com/sphinx-doc/sphinx/issues/4357>

²⁸¹⁴ <https://github.com/sphinx-doc/sphinx/issues/4334>

²⁸¹⁵ <https://github.com/sphinx-doc/sphinx/issues/4206>

²⁸¹⁶ <https://github.com/sphinx-doc/sphinx/issues/4231>

²⁸¹⁷ <https://github.com/sphinx-doc/sphinx/issues/4221>

²⁸¹⁸ <https://github.com/sphinx-doc/sphinx/issues/2298>

- [#4099](#)²⁸¹⁹: C++: properly link class reference to class from inside constructor
- [#4267](#)²⁸²⁰: PDF build broken by Unicode U+2116 NUMERO SIGN character
- [#4249](#)²⁸²¹: PDF output: Pygments error highlighting increases line spacing in code blocks
- [#1238](#)²⁸²²: Support `:emphasize-lines:` in PDF output
- [#4279](#)²⁸²³: Sphinx crashes with pickling error when run with multiple processes and remote image
- [#1421](#)²⁸²⁴: Respect the quiet flag in sphinx-quickstart
- [#4281](#)²⁸²⁵: Race conditions when creating output directory
- [#4315](#)²⁸²⁶: For PDF 'howto' documents, `latex_toplevel_sectioning='part'` generates `\chapter` commands
- [#4214](#)²⁸²⁷: Two todoclist directives break Sphinx 1.6.5
- Fix links to external option docs with intersphinx (refs: [#3769](#)²⁸²⁸)
- [#4091](#)²⁸²⁹: Private members not documented without `:undoc-members:`

14.111 Release 1.6.5 (released Oct 23, 2017)

Funzionalità aggiunte

- [#4107](#)²⁸³⁰: Make searchtools.js compatible with pre-Sphinx 1.5 templates
- [#4112](#)²⁸³¹: Don't override the `smart_quotes` setting if it was already set
- [#4125](#)²⁸³²: Display reference texts of original and translated passages on i18n warning message
- [#4147](#)²⁸³³: Include the exception when logging PO/MO file read/write

²⁸¹⁹ <https://github.com/sphinx-doc/sphinx/issues/4099>

²⁸²⁰ <https://github.com/sphinx-doc/sphinx/issues/4267>

²⁸²¹ <https://github.com/sphinx-doc/sphinx/issues/4249>

²⁸²² <https://github.com/sphinx-doc/sphinx/issues/1238>

²⁸²³ <https://github.com/sphinx-doc/sphinx/issues/4279>

²⁸²⁴ <https://github.com/sphinx-doc/sphinx/issues/1421>

²⁸²⁵ <https://github.com/sphinx-doc/sphinx/issues/4281>

²⁸²⁶ <https://github.com/sphinx-doc/sphinx/issues/4315>

²⁸²⁷ <https://github.com/sphinx-doc/sphinx/issues/4214>

²⁸²⁸ <https://github.com/sphinx-doc/sphinx/issues/3769>

²⁸²⁹ <https://github.com/sphinx-doc/sphinx/issues/4091>

²⁸³⁰ <https://github.com/sphinx-doc/sphinx/issues/4107>

²⁸³¹ <https://github.com/sphinx-doc/sphinx/issues/4112>

²⁸³² <https://github.com/sphinx-doc/sphinx/issues/4125>

²⁸³³ <https://github.com/sphinx-doc/sphinx/issues/4147>

Bugs fixed

- [#4085](#)²⁸³⁴: Failed PDF build from image in parsed-literal using `:align:` option
- [#4100](#)²⁸³⁵: Remove debug print from autodoc extension
- [#3987](#)²⁸³⁶: Changing theme from `alabaster` causes HTML build to fail
- [#4096](#)²⁸³⁷: C++, don't crash when using the wrong role type. Thanks to [mitya57](#).
- [#4070](#)²⁸³⁸, [#4111](#)²⁸³⁹: crashes when the warning message contains format strings (again)
- [#4108](#)²⁸⁴⁰: Search word highlighting breaks SVG images
- [#3692](#)²⁸⁴¹: Unable to build HTML if writing `.buildinfo` failed
- [#4152](#)²⁸⁴²: HTML writer crashes if a field list is placed on top of the document
- [#4063](#)²⁸⁴³: Sphinx crashes when labeling directive `.. todolist::`
- [#4134](#)²⁸⁴⁴: `[doc] docutils.conf` is not documented explicitly
- [#4169](#)²⁸⁴⁵: Chinese language doesn't trigger Chinese search automatically
- [#1020](#)²⁸⁴⁶: `ext.todo todolist` not linking to the page in `pdflatex`
- [#3965](#)²⁸⁴⁷: New quickstart generates wrong `SPHINXBUILD` in Makefile
- [#3739](#)²⁸⁴⁸: `:module:` option is ignored at content of pyobjects
- [#4149](#)²⁸⁴⁹: Documentation: Help choosing `latex_engine`
- [#4090](#)²⁸⁵⁰: `[doc] latex_additional_files` with extra LaTeX macros should not use `.tex` extension
- Failed to convert reST parser error to warning (refs: [#4132](#)²⁸⁵¹)

²⁸³⁴ <https://github.com/sphinx-doc/sphinx/issues/4085>

²⁸³⁵ <https://github.com/sphinx-doc/sphinx/issues/4100>

²⁸³⁶ <https://github.com/sphinx-doc/sphinx/issues/3987>

²⁸³⁷ <https://github.com/sphinx-doc/sphinx/issues/4096>

²⁸³⁸ <https://github.com/sphinx-doc/sphinx/issues/4070>

²⁸³⁹ <https://github.com/sphinx-doc/sphinx/issues/4111>

²⁸⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/4108>

²⁸⁴¹ <https://github.com/sphinx-doc/sphinx/issues/3692>

²⁸⁴² <https://github.com/sphinx-doc/sphinx/issues/4152>

²⁸⁴³ <https://github.com/sphinx-doc/sphinx/issues/4063>

²⁸⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/4134>

²⁸⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/4169>

²⁸⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/1020>

²⁸⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/3965>

²⁸⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/3739>

²⁸⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/4149>

²⁸⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/4090>

²⁸⁵¹ <https://github.com/sphinx-doc/sphinx/issues/4132>

14.112 Release 1.6.4 (released Sep 26, 2017)

Funzionalità aggiunte

- [#3926](#)²⁸⁵²: Add `autodoc_warningiserror` to suppress the behavior of `-W` option during importing target modules on autodoc

Bugs fixed

- [#3924](#)²⁸⁵³: docname lost after dynamically parsing RST in extension
- [#3946](#)²⁸⁵⁴: Typo in `sphinx.sty` (this was a bug with no effect in default context)
- **pep**
and `:rfc:` does not supports `default-role` directive (refs: [#3960](#)²⁸⁵⁵)
- [#3960](#)²⁸⁵⁶: `default_role = 'guilabel'` not functioning
- Missing `texinputs_win/Makefile` to be used in `latexpdf` builder on windows.
- [#4026](#)²⁸⁵⁷: nature: Fix macOS Safari scrollbar color
- [#3877](#)²⁸⁵⁸: Fix for C++ multiline signatures.
- [#4006](#)²⁸⁵⁹: Fix crash on parallel build
- [#3969](#)²⁸⁶⁰: private instance attributes causes `AttributeError`
- [#4041](#)²⁸⁶¹: C++, remove extra name linking in function pointers.
- [#4038](#)²⁸⁶²: C, add missing documentation of `member` role.
- [#4044](#)²⁸⁶³: An empty multicolumn cell causes extra row height in PDF output
- [#4049](#)²⁸⁶⁴: Fix typo in output of `sphinx-build -h`
- [#4062](#)²⁸⁶⁵: `hashlib.sha1()` must take bytes, not unicode on Python 3
- Avoid indent after index entries in latex (refs: [#4066](#)²⁸⁶⁶)

²⁸⁵² <https://github.com/sphinx-doc/sphinx/issues/3926>

²⁸⁵³ <https://github.com/sphinx-doc/sphinx/issues/3924>

²⁸⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/3946>

²⁸⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/3960>

²⁸⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/3960>

²⁸⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/4026>

²⁸⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/3877>

²⁸⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/4006>

²⁸⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/3969>

²⁸⁶¹ <https://github.com/sphinx-doc/sphinx/issues/4041>

²⁸⁶² <https://github.com/sphinx-doc/sphinx/issues/4038>

²⁸⁶³ <https://github.com/sphinx-doc/sphinx/issues/4044>

²⁸⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/4049>

²⁸⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/4062>

²⁸⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/4066>

- [#4070](#)²⁸⁶⁷: crashes when the warning message contains format strings
- [#4067](#)²⁸⁶⁸: Return non-zero exit status when make subprocess fails
- [#4055](#)²⁸⁶⁹: graphviz: the :align: option does not work for SVG output
- [#4055](#)²⁸⁷⁰: graphviz: the :align: center option does not work for latex output
- [#4051](#)²⁸⁷¹: warn() function for HTML theme outputs 'None' string

14.113 Release 1.6.3 (released Jul 02, 2017)

Funzionalità aggiunte

- latex: hint that code-block continues on next page (refs: [#3764](#)²⁸⁷², [#3792](#)²⁸⁷³)

Bugs fixed

- [#3821](#)²⁸⁷⁴: Failed to import sphinx.util.compat with Docutils 0.14rc1
- [#3829](#)²⁸⁷⁵: sphinx-quickstart template is incomplete regarding use of alabaster
- [#3772](#)²⁸⁷⁶: 'str object' has no attribute 'filename'
- Emit wrong warnings if citation label includes hyphens (refs: [#3565](#)²⁸⁷⁷)
- [#3858](#)²⁸⁷⁸: Some warnings are not colored when using --color option
- [#3775](#)²⁸⁷⁹: Remove unwanted whitespace in default template
- [#3835](#)²⁸⁸⁰: sphinx.ext.imgmath fails to convert SVG images if project directory name contains spaces
- [#3850](#)²⁸⁸¹: Fix color handling in make mode's help command
- [#3865](#)²⁸⁸²: use of self.env.warn in Sphinx extension fails

²⁸⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/4070>

²⁸⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/4067>

²⁸⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/4055>

²⁸⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/4055>

²⁸⁷¹ <https://github.com/sphinx-doc/sphinx/issues/4051>

²⁸⁷² <https://github.com/sphinx-doc/sphinx/issues/3764>

²⁸⁷³ <https://github.com/sphinx-doc/sphinx/issues/3792>

²⁸⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/3821>

²⁸⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/3829>

²⁸⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/3772>

²⁸⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/3565>

²⁸⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/3858>

²⁸⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/3775>

²⁸⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/3835>

²⁸⁸¹ <https://github.com/sphinx-doc/sphinx/issues/3850>

²⁸⁸² <https://github.com/sphinx-doc/sphinx/issues/3865>

- [#3824²⁸⁸³](#): production lists apply smart quotes transform since Sphinx 1.6.1
- latex: fix `\sphinxbfcode` swallows initial space of argument
- [#3878²⁸⁸⁴](#): Quotes in auto-documented class attributes should be straight quotes in PDF output
- [#3881²⁸⁸⁵](#): LaTeX figure floated to next page sometimes leaves extra vertical whitespace
- [#3885²⁸⁸⁶](#): duplicated footnotes raises `IndexError`
- [#3873²⁸⁸⁷](#): Failure of deprecation warning mechanism of `sphinx.util.compat.Directive`
- [#3874²⁸⁸⁸](#): Bogus warnings for "citation not referenced" for cross-file citations
- [#3860²⁸⁸⁹](#): Don't download images when builders not supported images
- [#3860²⁸⁹⁰](#): Remote image URIs without filename break builders not supported remote images
- [#3833²⁸⁹¹](#): command line messages are translated unintentionally with language setting.
- [#3840²⁸⁹²](#): make checking `epub_uid` strict
- [#3851²⁸⁹³](#), [#3706²⁸⁹⁴](#): Fix about box drawing characters for PDF output
- [#3900²⁸⁹⁵](#): autosummary could not find methods
- [#3902²⁸⁹⁶](#): Emit error if `latex_documents` contains non-unicode string in py2

14.114 Release 1.6.2 (released May 28, 2017)

Modifiche incompatibili

- [#3789²⁸⁹⁷](#): Do not require typing module for python>=3.5

²⁸⁸³ <https://github.com/sphinx-doc/sphinx/issues/3824>

²⁸⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/3878>

²⁸⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/3881>

²⁸⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/3885>

²⁸⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/3873>

²⁸⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/3874>

²⁸⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/3860>

²⁸⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/3860>

²⁸⁹¹ <https://github.com/sphinx-doc/sphinx/issues/3833>

²⁸⁹² <https://github.com/sphinx-doc/sphinx/issues/3840>

²⁸⁹³ <https://github.com/sphinx-doc/sphinx/issues/3851>

²⁸⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/3706>

²⁸⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/3900>

²⁸⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/3902>

²⁸⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/3789>

Bugs fixed

- [#3754](#)²⁸⁹⁸: HTML builder crashes if HTML theme appends own stylesheets
- [#3756](#)²⁸⁹⁹: epub: Entity 'mdash' not defined
- [#3758](#)²⁹⁰⁰: Sphinx crashed if logs are emitted in conf.py
- [#3755](#)²⁹⁰¹: incorrectly warns about dedent with literalinclude
- [#3742](#)²⁹⁰²: [RTD](#)²⁹⁰³ PDF builds of Sphinx own docs are missing an index entry in the bookmarks and table of contents. This is [rtfd/readthedocs.org`#2857](#)²⁹⁰⁴ <<https://github.com/rtfd/readthedocs.org/issues/2857>>`_ issue, a workaround is obtained using some extra LaTeX code in Sphinx's own conf.py
- [#3770](#)²⁹⁰⁵: Build fails when a "code-block" has the option emphasize-lines and the number indicated is higher than the number of lines
- [#3774](#)²⁹⁰⁶: Incremental HTML building broken when using citations
- [#3763](#)²⁹⁰⁷: got epubcheck validations error if epub_cover is set
- [#3779](#)²⁹⁰⁸: 'ImportError' in sphinx.ext.autodoc due to broken 'sys.meta_path'. Thanks to Tatiana Tereshchenko.
- [#3796](#)²⁹⁰⁹: env.resolve_references() crashes when non-document node given
- [#3803](#)²⁹¹⁰: Sphinx crashes with invalid PO files
- [#3791](#)²⁹¹¹: PDF "continued on next page" for long tables isn't internationalized
- [#3788](#)²⁹¹²: smartquotes emits warnings for unsupported languages
- [#3807](#)²⁹¹³: latex Makefile for make latexpdf is only for unixen
- [#3781](#)²⁹¹⁴: double hyphens in option directive are compiled as endashes
- [#3817](#)²⁹¹⁵: latex builder raises AttributeError

²⁸⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/3754>

²⁸⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/3756>

²⁹⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/3758>

²⁹⁰¹ <https://github.com/sphinx-doc/sphinx/issues/3755>

²⁹⁰² <https://github.com/sphinx-doc/sphinx/issues/3742>

²⁹⁰³ <https://readthedocs.org/>

²⁹⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/2857>

²⁹⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/3770>

²⁹⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/3774>

²⁹⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/3763>

²⁹⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/3779>

²⁹⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/3796>

²⁹¹⁰ <https://github.com/sphinx-doc/sphinx/issues/3803>

²⁹¹¹ <https://github.com/sphinx-doc/sphinx/issues/3791>

²⁹¹² <https://github.com/sphinx-doc/sphinx/issues/3788>

²⁹¹³ <https://github.com/sphinx-doc/sphinx/issues/3807>

²⁹¹⁴ <https://github.com/sphinx-doc/sphinx/issues/3781>

²⁹¹⁵ <https://github.com/sphinx-doc/sphinx/issues/3817>

14.115 Release 1.6.1 (released May 16, 2017)

Dipendenze

1.6b1

- (updated) latex output is tested with Ubuntu trusty's texlive packages (Feb. 2014) and earlier tex installations may not be fully compliant, particularly regarding Unicode engines xelatex and lualatex
- (added) latexmk is required for make latexpdf on GNU/Linux and Mac OS X (refs: [#3082](#)²⁹¹⁶)

Modifiche incompatibili

1.6b1

- [#1061](#)²⁹¹⁷, [#2336](#)²⁹¹⁸, [#3235](#)²⁹¹⁹: Now generation of autosummary doesn't contain imported members by default. Thanks to Luc Saffre.
- LaTeX `\includegraphics` command isn't overloaded: only `\sphinxincludegraphics` has the custom code to fit image to available width if oversized.
- The subclasses of `sphinx.domains.Index` should override `generate()` method. The default implementation raises `NotImplementedError`
- LaTeX positioned long tables horizontally centered, and short ones flushed left (no text flow around table.) The position now defaults to center in both cases, and it will obey Docutils 0.13 `:align:` option (refs [#3415](#)²⁹²⁰, [#3377](#)²⁹²¹)
- option directive also allows all punctuations for the option name (refs: [#3366](#)²⁹²²)
- [#3413](#)²⁹²³: if `literalinclude`'s `:start-after:` is used, make `:lines:` relative (refs [#3412](#)²⁹²⁴)
- `literalinclude` directive does not allow the combination of `:diff:` option and other options (refs: [#3416](#)²⁹²⁵)
- LuaLaTeX engine uses `fontspec` like XeLaTeX. It is advised `latex_engine = 'lualatex'` be used only on up-to-date TeX installs (refs [#3070](#)²⁹²⁶, [#3466](#)²⁹²⁷)

²⁹¹⁶ <https://github.com/sphinx-doc/sphinx/issues/3082>

²⁹¹⁷ <https://github.com/sphinx-doc/sphinx/issues/1061>

²⁹¹⁸ <https://github.com/sphinx-doc/sphinx/issues/2336>

²⁹¹⁹ <https://github.com/sphinx-doc/sphinx/issues/3235>

²⁹²⁰ <https://github.com/sphinx-doc/sphinx/issues/3415>

²⁹²¹ <https://github.com/sphinx-doc/sphinx/issues/3377>

²⁹²² <https://github.com/sphinx-doc/sphinx/issues/3366>

²⁹²³ <https://github.com/sphinx-doc/sphinx/issues/3413>

²⁹²⁴ <https://github.com/sphinx-doc/sphinx/issues/3412>

²⁹²⁵ <https://github.com/sphinx-doc/sphinx/issues/3416>

²⁹²⁶ <https://github.com/sphinx-doc/sphinx/issues/3070>

²⁹²⁷ <https://github.com/sphinx-doc/sphinx/issues/3466>

- `latex_keep_old_macro_names` default value has been changed from `True` to `False`. This means that some LaTeX macros for styling are by default defined only with `\sphinx..` prefixed names. (refs: [#3429](#)²⁹²⁸)
- Footer "Continued on next page" of LaTeX `longtable`'s now not framed (refs: [#3497](#)²⁹²⁹)
- [#3529](#)²⁹³⁰: The arguments of `BuildEnvironment.__init__` is changed
- [#3082](#)²⁹³¹: Use `latexmk` for pdf (and dvi) targets (Unix-like platforms only)
- [#3558](#)²⁹³²: Emit warnings if footnotes and citations are not referenced. The warnings can be suppressed by `suppress_warnings`.
- `latex` made available (non documented) colour macros from a file distributed with `pdftex` engine for Plain TeX. This is removed in order to provide better support for multiple TeX engines. Only interface from `color` or `xcolor` packages should be used by extensions of Sphinx latex writer. (refs [#3550](#)²⁹³³)
- `Builder.env` is not filled at instantiation
- [#3594](#)²⁹³⁴: LaTeX: single raw directive has been considered as block level element
- [#3639](#)²⁹³⁵: If `html_experimental_html5_writer` is available, epub builder use it by default.
- `Sphinx.add_source_parser()` raises an error if duplicated

1.6b2

- [#3345](#)²⁹³⁶: Replace the custom `smartypants` code with Docutils' `smart_quotes`. Thanks to Dmitry Shachnev, and to Günter Milde at Docutils.

1.6b3

- LaTeX package `eqparbox` is not used and not loaded by Sphinx anymore
- LaTeX package `multitrow` is not used and not loaded by Sphinx anymore
- Add line numbers to citation data in std domain

1.6 final

- LaTeX package `threeparttable` is not used and not loaded by Sphinx anymore (refs [#3686](#)²⁹³⁷, [#3532](#)²⁹³⁸, [#3377](#)²⁹³⁹)

²⁹²⁸ <https://github.com/sphinx-doc/sphinx/issues/3429>²⁹²⁹ <https://github.com/sphinx-doc/sphinx/issues/3497>²⁹³⁰ <https://github.com/sphinx-doc/sphinx/issues/3529>²⁹³¹ <https://github.com/sphinx-doc/sphinx/issues/3082>²⁹³² <https://github.com/sphinx-doc/sphinx/issues/3558>²⁹³³ <https://github.com/sphinx-doc/sphinx/issues/3550>²⁹³⁴ <https://github.com/sphinx-doc/sphinx/issues/3594>²⁹³⁵ <https://github.com/sphinx-doc/sphinx/issues/3639>²⁹³⁶ <https://github.com/sphinx-doc/sphinx/issues/3345>²⁹³⁷ <https://github.com/sphinx-doc/sphinx/issues/3686>²⁹³⁸ <https://github.com/sphinx-doc/sphinx/issues/3532>²⁹³⁹ <https://github.com/sphinx-doc/sphinx/issues/3377>

Features removed

- Configuration variables
 - `epub3_contributor`
 - `epub3_description`
 - `epub3_page_progression_direction`
 - `html_translator_class`
 - `html_use_modindex`
 - `latex_font_size`
 - `latex_paper_size`
 - `latex_preamble`
 - `latex_use_modindex`
 - `latex_use_parts`
- `termsep` node
- `defindex.html` template
- LDML format support in `today`, `today_fmt` and `html_last_updated_fmt`
- `:inline:` option for the directives of `sphinx.ext.graphviz` extension
- `sphinx.ext.pngmath` extension
- `sphinx.util.compat.make_admonition()`

Funzionalità aggiunte

1.6b1

- [#3136](https://github.com/sphinx-doc/sphinx/issues/3136)²⁹⁴⁰: Add `:name:` option to the directives in `sphinx.ext.graphviz`
- [#2336](https://github.com/sphinx-doc/sphinx/issues/2336)²⁹⁴¹: Add `imported_members` option to `sphinx-autogen` command to document imported members.
- C++, add `:tparam-line-spec:` option to templated declarations. When specified, each template parameter will be rendered on a separate line.
- [#3359](https://github.com/sphinx-doc/sphinx/issues/3359)²⁹⁴²: Allow `sphinx.js` in a user locale dir to override `sphinx.js` from Sphinx
- [#3303](https://github.com/sphinx-doc/sphinx/issues/3303)²⁹⁴³: Add `:pyversion:` option to the `doctest` directive.

²⁹⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/3136>

²⁹⁴¹ <https://github.com/sphinx-doc/sphinx/issues/2336>

²⁹⁴² <https://github.com/sphinx-doc/sphinx/issues/3359>

²⁹⁴³ <https://github.com/sphinx-doc/sphinx/issues/3303>

- [#3378](#)²⁹⁴⁴: (latex) support for `:widths:` option of table directives (refs: [#3379](#)²⁹⁴⁵, [#3381](#)²⁹⁴⁶)
- [#3402](#)²⁹⁴⁷: Allow to suppress "download file not readable" warnings using `suppress_warnings`.
- [#3377](#)²⁹⁴⁸: latex: Add support for Docutils 0.13 `:align:` option for tables (but does not implement text flow around table).
- latex: footnotes from inside tables are hyperlinked (except from captions or headers) (refs: [#3422](#)²⁹⁴⁹)
- Emit warning if over dedent has detected on `literalinclude` directive (refs: [#3416](#)²⁹⁵⁰)
- Use for LuaLaTeX same default settings as for XeLaTeX (i.e. `fontspec` and `polyglossia`). (refs: [#3070](#)²⁹⁵¹, [#3466](#)²⁹⁵²)
- Make 'extraclassoptions' key of `latex_elements` public (refs [#3480](#)²⁹⁵³)
- [#3463](#)²⁹⁵⁴: Add warning messages for required EPUB3 metadata. Add default value to `epub_description` to avoid warning like other settings.
- [#3476](#)²⁹⁵⁵: `setuptools`: Support multiple builders
- latex: merged cells in LaTeX tables allow code-blocks, lists, blockquotes... as do normal cells (refs: [#3435](#)²⁹⁵⁶)
- HTML builder uses experimental HTML5 writer if `html_experimental_html5_writer` is True and Docutils 0.13 or later is installed.
- LaTeX macros to customize space before and after tables in PDF output (refs [#3504](#)²⁹⁵⁷)
- [#3348](#)²⁹⁵⁸: Show decorators in `literalinclude` and `viewcode` directives
- [#3108](#)²⁹⁵⁹: Show warning if `:start-at:` and other `literalinclude` options does not match to the text
- [#3609](#)²⁹⁶⁰: Allow to suppress "duplicate citation" warnings using `suppress_warnings`
- [#2803](#)²⁹⁶¹: Discovery of builders by entry point

²⁹⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/3378>

²⁹⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/3379>

²⁹⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/3381>

²⁹⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/3402>

²⁹⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/3377>

²⁹⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/3422>

²⁹⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/3416>

²⁹⁵¹ <https://github.com/sphinx-doc/sphinx/issues/3070>

²⁹⁵² <https://github.com/sphinx-doc/sphinx/issues/3466>

²⁹⁵³ <https://github.com/sphinx-doc/sphinx/issues/3480>

²⁹⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/3463>

²⁹⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/3476>

²⁹⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/3435>

²⁹⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/3504>

²⁹⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/3348>

²⁹⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/3108>

²⁹⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/3609>

²⁹⁶¹ <https://github.com/sphinx-doc/sphinx/issues/2803>

- [#1764](#)²⁹⁶², [#1676](#)²⁹⁶³: Allow setting 'rel' and 'title' attributes for stylesheets
- [#3589](#)²⁹⁶⁴: Support remote images on non-HTML builders
- [#3589](#)²⁹⁶⁵: Support images in Data URI on non-HTML builders
- [#2961](#)²⁹⁶⁶: improve `autodoc_mock_imports`. Now the config value only requires to declare the top-level modules that should be mocked. Thanks to Robin Jarry.
- [#3449](#)²⁹⁶⁷: On py3, autodoc use `inspect.signature` for more accurate signature calculation. Thanks to Nathaniel J. Smith.
- [#3641](#)²⁹⁶⁸: Epub theme supports HTML structures that are generated by HTML5 writer.
- [#3644](#)²⁹⁶⁹ autodoc uses `inspect` instead of checking types. Thanks to Jeroen Demeyer.
- Add a new extension; `sphinx.ext.imgconverter`. It converts images in the document to appropriate format for builders
- latex: Use templates to render tables (refs [#3389](#)²⁹⁷⁰, [2a37b0e](#))

1.6b2

- `LATEXMKOPTS` variable for the Makefile in `$BUILDDIR/latex` to pass options to `latexmk` when executing `make latexpdf` (refs [#3695](#)²⁹⁷¹, [#3720](#)²⁹⁷²)
- Add a new event `env-check-consistency` to check consistency to extensions
- Add `Domain.check_consistency()` to check consistency

Bugs fixed

1.6b1

- `literalinclude` directive expands tabs after dedent-ing (refs: [#3416](#)²⁹⁷³)
- [#1574](#)²⁹⁷⁴: Paragraphs in table cell doesn't work in Latex output
- [#3288](#)²⁹⁷⁵: Table with merged headers not wrapping text
- [#3491](#)²⁹⁷⁶: Inconsistent vertical space around table and longtable in PDF

²⁹⁶² <https://github.com/sphinx-doc/sphinx/issues/1764>

²⁹⁶³ <https://github.com/sphinx-doc/sphinx/issues/1676>

²⁹⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/3589>

²⁹⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/3589>

²⁹⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/2961>

²⁹⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/3449>

²⁹⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/3641>

²⁹⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/3644>

²⁹⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/3389>

²⁹⁷¹ <https://github.com/sphinx-doc/sphinx/issues/3695>

²⁹⁷² <https://github.com/sphinx-doc/sphinx/issues/3720>

²⁹⁷³ <https://github.com/sphinx-doc/sphinx/issues/3416>

²⁹⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/1574>

²⁹⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/3288>

²⁹⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/3491>

- [#3506](#)²⁹⁷⁷: Depart functions for all admonitions in HTML writer now properly pass node to `depart_admonition`.
- [#2693](#)²⁹⁷⁸: Sphinx latex style file wrongly inhibits colours for section headings for latex+dvi(ps,pdf,pdfmx)
- C++, properly look up any references.
- [#3624](#)²⁹⁷⁹: sphinx.ext.intersphinx couldn't load inventories compressed with gzip
- [#3551](#)²⁹⁸⁰: PDF information dictionary is lacking author and title data
- [#3351](#)²⁹⁸¹: intersphinx does not refers context like `py:module`, `py:class` and so on.
- Fail to load template file if the parent template is archived

1.6b2

- [#3661](#)²⁹⁸²: sphinx-build crashes on parallel build
- [#3669](#)²⁹⁸³: gettext builder fails with "ValueError: substring not found"
- [#3660](#)²⁹⁸⁴: Sphinx always depends on sphinxcontrib-websupport and its dependencies
- [#3472](#)²⁹⁸⁵: smart quotes getting wrong in latex (at least with list of strings via `autoattribute`) (refs: [#3345](#)²⁹⁸⁶, [#3666](#)²⁹⁸⁷)

1.6b3

- [#3588](#)²⁹⁸⁸: No compact (p tag) html output in the i18n document build even when `html_compact_lists` is True.
- The `make latexpdf` from 1.6b1 (for GNU/Linux and Mac OS, using `latexmk`) aborted earlier in case of LaTeX errors than was the case with 1.5 series, due to hard-coded usage of `--halt-on-error` option (refs [#3695](#)²⁹⁸⁹)
- [#3683](#)²⁹⁹⁰: sphinx.websupport module is not provided by default
- [#3683](#)²⁹⁹¹: Failed to build document if `builder.css_file.insert()` is called
- [#3714](#)²⁹⁹²: viewcode extension not taking `highlight_code='none'` in account

²⁹⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/3506>

²⁹⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/2693>

²⁹⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/3624>

²⁹⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/3551>

²⁹⁸¹ <https://github.com/sphinx-doc/sphinx/issues/3351>

²⁹⁸² <https://github.com/sphinx-doc/sphinx/issues/3661>

²⁹⁸³ <https://github.com/sphinx-doc/sphinx/issues/3669>

²⁹⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/3660>

²⁹⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/3472>

²⁹⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/3345>

²⁹⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/3666>

²⁹⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/3588>

²⁹⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/3695>

²⁹⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/3683>

²⁹⁹¹ <https://github.com/sphinx-doc/sphinx/issues/3683>

²⁹⁹² <https://github.com/sphinx-doc/sphinx/issues/3714>

- [#3698](#)²⁹⁹³: Moving `:doc:` to std domain broke backwards compatibility
- [#3633](#)²⁹⁹⁴: misdetect unreferenced citations

1.6 final

- LaTeX tables do not allow multiple paragraphs in a header cell
- LATEXOPTS is not passed over correctly to pdflatex since 1.6b3
- [#3532](#)²⁹⁹⁵: Figure or literal block captions in cells of short tables cause havoc in PDF output
- Fix: in PDF captions of tables are rendered differently whether table is of longtable class or not (refs [#3686](#)²⁹⁹⁶)
- [#3725](#)²⁹⁹⁷: Todo looks different from note in LaTeX output
- [#3479](#)²⁹⁹⁸: stub-columns have no effect in LaTeX output
- [#3738](#)²⁹⁹⁹: Nonsensical code in theming.py
- [#3746](#)³⁰⁰⁰: PDF builds fail with latexmk 4.48 or earlier due to undefined options `-pdfxe` and `-pdflua`

Deprecato

1.6b1

- `sphinx.util.compat.Directive` class is now deprecated. Please use instead `docutils.parsers.rst.Directive`
- `sphinx.util.compat.docutils_version` is now deprecated
- [#2367](#)³⁰⁰¹: `Sphinx.warn()`, `Sphinx.info()` and other logging methods are now deprecated. Please use `sphinx.util.logging` (*API per i Log*) instead.
- [#3318](#)³⁰⁰²: `notice` is now deprecated as LaTeX environment name and will be removed at Sphinx 1.7. Extension authors please use `sphinxadmonition` instead (as Sphinx does since 1.5.)
- `Sphinx.status_iterator()` and `Sphinx.old_status_iterator()` is now deprecated. Please use `sphinx.util:status_iterator()` instead.
- `Sphinx._directive_helper()` is deprecated. Please use `sphinx.util.docutils.directive_helper()` instead.

²⁹⁹³ <https://github.com/sphinx-doc/sphinx/issues/3698>

²⁹⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/3633>

²⁹⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/3532>

²⁹⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/3686>

²⁹⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/3725>

²⁹⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/3479>

²⁹⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/3738>

³⁰⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/3746>

³⁰⁰¹ <https://github.com/sphinx-doc/sphinx/issues/2367>

³⁰⁰² <https://github.com/sphinx-doc/sphinx/issues/3318>

- `BuildEnvironment.set_warnfunc()` is now deprecated
- Following methods of `BuildEnvironment` is now deprecated.
 - `BuildEnvironment.note_toc_tree()`
 - `BuildEnvironment.get_toc_for()`
 - `BuildEnvironment.get_toc_tree_for()`
 - `BuildEnvironment.create_index()`

Please use `sphinx.environment.adapters` modules instead.

- latex package `footnote` is not loaded anymore by its bundled replacement `footnotehyper-sphinx`. The redefined macros keep the same names as in the original package.
- #3429³⁰⁰³: deprecate config setting `latex_keep_old_macro_names`. It will be removed at 1.7, and already its default value has changed from `True` to `False`.
- #3221³⁰⁰⁴: `epub2` builder is deprecated
- #3254³⁰⁰⁵: `sphinx.websupport` is now separated into independent package; `sphinxcontrib-websupport`. `sphinx.websupport` will be removed in Sphinx 2.0.
- #3628³⁰⁰⁶: `sphinx_themes` entry_point is deprecated. Please use `sphinx.html_themes` instead.

1.6b2

- #3662³⁰⁰⁷: `builder.css_files` is deprecated. Please use `add_stylesheet()` API instead.

1.6 final

- LaTeX `\sphinxstylethead` is deprecated at 1.6 and will be removed at 1.7. Please move customization into new macro `\sphinxstyletheadfamily`.

Testing

1.6 final

- #3458³⁰⁰⁸: Add `sphinx.testing` (experimental)

³⁰⁰³ <https://github.com/sphinx-doc/sphinx/issues/3429>

³⁰⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/3221>

³⁰⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/3254>

³⁰⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/3628>

³⁰⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/3662>

³⁰⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/3458>

14.116 Release 1.6 (unreleased)

- not released (because of package script error)

14.117 Release 1.5.6 (released May 15, 2017)

Bugs fixed

- [#3614](https://github.com/sphinx-doc/sphinx/issues/3614)³⁰⁰⁹: Sphinx crashes with requests-2.5.0
- [#3618](https://github.com/sphinx-doc/sphinx/issues/3618)³⁰¹⁰: autodoc crashes with tupled arguments
- [#3664](https://github.com/sphinx-doc/sphinx/issues/3664)³⁰¹¹: No space after the bullet in items of a latex list produced by Sphinx
- [#3657](https://github.com/sphinx-doc/sphinx/issues/3657)³⁰¹²: EPUB builder crashes if a document starting with genindex exists
- [#3588](https://github.com/sphinx-doc/sphinx/issues/3588)³⁰¹³: No compact (p tag) html output in the i18n document build even when `html_compact_lists` is True.
- [#3685](https://github.com/sphinx-doc/sphinx/issues/3685)³⁰¹⁴: AttributeError when using 3rd party domains
- [#3702](https://github.com/sphinx-doc/sphinx/issues/3702)³⁰¹⁵: LaTeX writer styles figure legends with a hard-coded `\small`
- [#3708](https://github.com/sphinx-doc/sphinx/issues/3708)³⁰¹⁶: LaTeX writer allows irc scheme
- [#3717](https://github.com/sphinx-doc/sphinx/issues/3717)³⁰¹⁷: Stop enforcing that favicon's must be .ico
- [#3731](https://github.com/sphinx-doc/sphinx/issues/3731)³⁰¹⁸, [#3732](https://github.com/sphinx-doc/sphinx/issues/3732)³⁰¹⁹: Protect isenumclass predicate against non-class arguments
- [#3320](https://github.com/sphinx-doc/sphinx/issues/3320)³⁰²⁰: Warning about reference target not being found for container types
- Misspelled ARCHIVEPREFIX in Makefile for latex build repertory

³⁰⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/3614>

³⁰¹⁰ <https://github.com/sphinx-doc/sphinx/issues/3618>

³⁰¹¹ <https://github.com/sphinx-doc/sphinx/issues/3664>

³⁰¹² <https://github.com/sphinx-doc/sphinx/issues/3657>

³⁰¹³ <https://github.com/sphinx-doc/sphinx/issues/3588>

³⁰¹⁴ <https://github.com/sphinx-doc/sphinx/issues/3685>

³⁰¹⁵ <https://github.com/sphinx-doc/sphinx/issues/3702>

³⁰¹⁶ <https://github.com/sphinx-doc/sphinx/issues/3708>

³⁰¹⁷ <https://github.com/sphinx-doc/sphinx/issues/3717>

³⁰¹⁸ <https://github.com/sphinx-doc/sphinx/issues/3731>

³⁰¹⁹ <https://github.com/sphinx-doc/sphinx/issues/3732>

³⁰²⁰ <https://github.com/sphinx-doc/sphinx/issues/3320>

14.118 Release 1.5.5 (released Apr 03, 2017)

Bugs fixed

- [#3597](#)³⁰²¹: python domain raises UnboundLocalError if invalid name given
- [#3599](#)³⁰²²: Move to new MathJax CDN

14.119 Release 1.5.4 (released Apr 02, 2017)

Funzionalità aggiunte

- [#3470](#)³⁰²³: Make genindex support all kinds of letters, not only Latin ones

Bugs fixed

- [#3445](#)³⁰²⁴: setting 'inputenc' key to `\usepackage[utf8x]{inputenc}` leads to failed PDF build
- EPUB file has duplicated `nav.xhtml` link in `content.opf` except first time build
- [#3488](#)³⁰²⁵: `objects.inv` has broken when `release` or `version` contain return code
- [#2073](#)³⁰²⁶, [#3443](#)³⁰²⁷, [#3490](#)³⁰²⁸: gettext builder that writes pot files unless the content are same without creation date. Thanks to Yoshiki Shibukawa.
- [#3487](#)³⁰²⁹: intersphinx: failed to refer options
- [#3496](#)³⁰³⁰: latex longtable's last column may be much wider than its contents
- [#3507](#)³⁰³¹: wrong quotes in latex output for `productionlist` directive
- [#3533](#)³⁰³²: Moving from Sphinx 1.3.1 to 1.5.3 breaks LaTeX compilation of links rendered as code

³⁰²¹ <https://github.com/sphinx-doc/sphinx/issues/3597>

³⁰²² <https://github.com/sphinx-doc/sphinx/issues/3599>

³⁰²³ <https://github.com/sphinx-doc/sphinx/issues/3470>

³⁰²⁴ <https://github.com/sphinx-doc/sphinx/issues/3445>

³⁰²⁵ <https://github.com/sphinx-doc/sphinx/issues/3488>

³⁰²⁶ <https://github.com/sphinx-doc/sphinx/issues/2073>

³⁰²⁷ <https://github.com/sphinx-doc/sphinx/issues/3443>

³⁰²⁸ <https://github.com/sphinx-doc/sphinx/issues/3490>

³⁰²⁹ <https://github.com/sphinx-doc/sphinx/issues/3487>

³⁰³⁰ <https://github.com/sphinx-doc/sphinx/issues/3496>

³⁰³¹ <https://github.com/sphinx-doc/sphinx/issues/3507>

³⁰³² <https://github.com/sphinx-doc/sphinx/issues/3533>

- [#2665](#)³⁰³³, [#2607](#)³⁰³⁴: Link names in C++ docfields, and make it possible for other domains.
- [#3542](#)³⁰³⁵: C++, fix parsing error of non-type template argument with template.
- [#3065](#)³⁰³⁶, [#3520](#)³⁰³⁷: python domain fails to recognize nested class
- [#3575](#)³⁰³⁸: Problems with pdflatex in a Turkish document built with Sphinx has reappeared (refs [#2997](#)³⁰³⁹, [#2397](#)³⁰⁴⁰)
- [#3577](#)³⁰⁴¹: Fix intersphinx debug tool
- A LaTeX command such as `\large` inserted in the title items of `latex_documents` causes failed PDF build (refs [#3551](#)³⁰⁴², [#3567](#)³⁰⁴³)

14.120 Release 1.5.3 (released Feb 26, 2017)

Funzionalità aggiunte

- Support requests-2.0.0 (experimental) (refs: [#3367](#)³⁰⁴⁴)
- (latex) PDF page margin dimensions may be customized (refs: [#3387](#)³⁰⁴⁵)
- `literalinclude` directive allows combination of `:pyobject:` and `:lines:` options (refs: [#3416](#)³⁰⁴⁶)
- [#3400](#)³⁰⁴⁷: make-mode doesn't use subprocess on building docs

³⁰³³ <https://github.com/sphinx-doc/sphinx/issues/2665>

³⁰³⁴ <https://github.com/sphinx-doc/sphinx/issues/2607>

³⁰³⁵ <https://github.com/sphinx-doc/sphinx/issues/3542>

³⁰³⁶ <https://github.com/sphinx-doc/sphinx/issues/3065>

³⁰³⁷ <https://github.com/sphinx-doc/sphinx/issues/3520>

³⁰³⁸ <https://github.com/sphinx-doc/sphinx/issues/3575>

³⁰³⁹ <https://github.com/sphinx-doc/sphinx/issues/2997>

³⁰⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/2397>

³⁰⁴¹ <https://github.com/sphinx-doc/sphinx/issues/3577>

³⁰⁴² <https://github.com/sphinx-doc/sphinx/issues/3551>

³⁰⁴³ <https://github.com/sphinx-doc/sphinx/issues/3567>

³⁰⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/3367>

³⁰⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/3387>

³⁰⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/3416>

³⁰⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/3400>

Bugs fixed

- #3370³⁰⁴⁸: the caption of code-block is not picked up for translation
- LaTeX: *release* is not escaped (refs: #3362³⁰⁴⁹)
- #3364³⁰⁵⁰: sphinx-quickstart prompts overflow on Console with 80 chars width
- since 1.5, PDF's TOC and bookmarks lack an entry for general Index (refs: #3383³⁰⁵¹)
- #3392³⁰⁵²: 'releasename' in *latex_elements* is not working
- #3356³⁰⁵³: Page layout for Japanese 'manual' docclass has a shorter text area
- #3394³⁰⁵⁴: When 'pointsize' is not 10pt, Japanese 'manual' document gets wrong PDF page dimensions
- #3399³⁰⁵⁵: quickstart: conf.py was not overwritten by template
- #3366³⁰⁵⁶: option directive does not allow punctuations
- #3410³⁰⁵⁷: return code in *release* breaks html search
- #3427³⁰⁵⁸: autodoc: memory addresses are not stripped on Windows
- #3428³⁰⁵⁹: xetex build tests fail due to fontspec v2.6 defining \strong
- #3349³⁰⁶⁰: Result of `IndexBuilder.load()` is broken
- #3450³⁰⁶¹: is appeared in EPUB docs
- #3418³⁰⁶²: Search button is misaligned in nature and pyramid theme
- #3421³⁰⁶³: Could not translate a caption of tables
- #3552³⁰⁶⁴: linkcheck raises UnboundLocalError

³⁰⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/3370>

³⁰⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/3362>

³⁰⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/3364>

³⁰⁵¹ <https://github.com/sphinx-doc/sphinx/issues/3383>

³⁰⁵² <https://github.com/sphinx-doc/sphinx/issues/3392>

³⁰⁵³ <https://github.com/sphinx-doc/sphinx/issues/3356>

³⁰⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/3394>

³⁰⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/3399>

³⁰⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/3366>

³⁰⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/3410>

³⁰⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/3427>

³⁰⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/3428>

³⁰⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/3349>

³⁰⁶¹ <https://github.com/sphinx-doc/sphinx/issues/3450>

³⁰⁶² <https://github.com/sphinx-doc/sphinx/issues/3418>

³⁰⁶³ <https://github.com/sphinx-doc/sphinx/issues/3421>

³⁰⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/3552>

14.121 Release 1.5.2 (released Jan 22, 2017)

Modifiche incompatibili

- Dependency requirement updates: requests 2.4.0 or above (refs: [#3268](#)³⁰⁶⁵, [#3310](#)³⁰⁶⁶)

Funzionalità aggiunte

- [#3241](#)³⁰⁶⁷: emit latex warning if buggy titlesec (ref [#3210](#)³⁰⁶⁸)
- [#3194](#)³⁰⁶⁹: Refer the \$MAKE environment variable to determine make command
- Emit warning for nested numbered toctrees (refs: [#3142](#)³⁰⁷⁰)
- [#978](#)³⁰⁷¹: `intersphinx_mapping` also allows a list as a parameter
- [#3340](#)³⁰⁷²: (LaTeX) long lines in `parsed-literal`³⁰⁷³ are wrapped like in `code-block`, inline math and footnotes are fully functional.

Bugs fixed

- [#3246](#)³⁰⁷⁴: xapian search adapter crashes
- [#3253](#)³⁰⁷⁵: In Py2 environment, building another locale with a non-captioned toctree produces None captions
- [#185](#)³⁰⁷⁶: References to section title including raw node has broken
- [#3255](#)³⁰⁷⁷: In Py3.4 environment, autodoc doesn't support documentation for attributes of Enum class correctly.
- [#3261](#)³⁰⁷⁸: `latex_use_parts` makes Sphinx crash
- The warning type `misc.highlighting_failure` does not work
- [#3294](#)³⁰⁷⁹: `add_latex_package()` make crashes non-LaTeX builders

³⁰⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/3268>

³⁰⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/3310>

³⁰⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/3241>

³⁰⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/3210>

³⁰⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/3194>

³⁰⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/3142>

³⁰⁷¹ <https://github.com/sphinx-doc/sphinx/issues/978>

³⁰⁷² <https://github.com/sphinx-doc/sphinx/issues/3340>

³⁰⁷³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#parsed-literal>

³⁰⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/3246>

³⁰⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/3253>

³⁰⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/185>

³⁰⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/3255>

³⁰⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/3261>

³⁰⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/3294>

- The caption of table are rendered as invalid HTML (refs: #3287³⁰⁸⁰)
- #3268³⁰⁸¹: Sphinx crashes with requests package from Debian jessie
- #3284³⁰⁸²: Sphinx crashes on parallel build with an extension which raises unserializable exception
- #3315³⁰⁸³: Bibliography crashes on latex build with docclass 'memoir'
- #3328³⁰⁸⁴: Could not refer rubric implicitly
- #3329³⁰⁸⁵: emit warnings if po file is invalid and can't read it. Also writing mo
- #3337³⁰⁸⁶: Ugly rendering of definition list term's classifier
- #3335³⁰⁸⁷: gettext does not extract field_name of a field in a field_list
- #2952³⁰⁸⁸: C++, fix refs to operator() functions.
- Fix Unicode super- and subscript digits in *code-block* and parsed-literal LaTeX output (ref #3342³⁰⁸⁹)
- LaTeX writer: leave " character inside parsed-literal as is (ref #3341³⁰⁹⁰)
- #3234³⁰⁹¹: intersphinx failed for encoded inventories
- #3158³⁰⁹²: too much space after captions in PDF output
- #3317³⁰⁹³: An URL in parsed-literal contents gets wrongly rendered in PDF if with hyphen
- LaTeX crash if the filename of an image inserted in parsed-literal via a substitution contains an hyphen (ref #3340³⁰⁹⁴)
- LaTeX rendering of inserted footnotes in parsed-literal is wrong (ref #3340³⁰⁹⁵)
- Inline math in parsed-literal is not rendered well by LaTeX (ref #3340³⁰⁹⁶)
- #3308³⁰⁹⁷: Parsed-literals don't wrap very long lines with pdf builder (ref #3340³⁰⁹⁸)

³⁰⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/3287>

³⁰⁸¹ <https://github.com/sphinx-doc/sphinx/issues/3268>

³⁰⁸² <https://github.com/sphinx-doc/sphinx/issues/3284>

³⁰⁸³ <https://github.com/sphinx-doc/sphinx/issues/3315>

³⁰⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/3328>

³⁰⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/3329>

³⁰⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/3337>

³⁰⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/3335>

³⁰⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/2952>

³⁰⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/3342>

³⁰⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/3341>

³⁰⁹¹ <https://github.com/sphinx-doc/sphinx/issues/3234>

³⁰⁹² <https://github.com/sphinx-doc/sphinx/issues/3158>

³⁰⁹³ <https://github.com/sphinx-doc/sphinx/issues/3317>

³⁰⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/3340>

³⁰⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/3340>

³⁰⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/3340>

³⁰⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/3308>

³⁰⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/3340>

- [#3295](#)³⁰⁹⁹: Could not import extension sphinx.builders.linkcheck
- [#3285](#)³¹⁰⁰: autosummary: asterisks are escaped twice
- LaTeX, pass dvipdfm option to geometry package for Japanese documents (ref [#3363](#)³¹⁰¹)
- Fix parselinenos() could not parse left half open range (cf. "-4")

14.122 Release 1.5.1 (released Dec 13, 2016)

Funzionalità aggiunte

- [#3214](#)³¹⁰²: Allow to suppress "unknown mimetype" warnings from epub builder using `suppress_warnings`.

Bugs fixed

- [#3195](#)³¹⁰³: Can not build in parallel
- [#3198](#)³¹⁰⁴: AttributeError is raised when toctree has 'self'
- [#3211](#)³¹⁰⁵: Remove untranslated Sphinx locale catalogs (it was covered by untranslated it_IT)
- [#3212](#)³¹⁰⁶: HTML Builders crashes with Docutils 0.13
- [#3207](#)³¹⁰⁷: more latex problems with references inside parsed-literal directive (`\DUrole`)
- [#3205](#)³¹⁰⁸: sphinx.util.requests crashes with old pyOpenSSL (< 0.14)
- [#3220](#)³¹⁰⁹: KeyError when having a duplicate citation
- [#3200](#)³¹¹⁰: LaTeX: xref inside desc_name not allowed
- [#3228](#)³¹¹¹: build_sphinx command crashes when missing dependency
- [#2469](#)³¹¹²: Ignore updates of catalog files for gettext builder. Thanks to Hiroshi Ohkubo.

³⁰⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/3295>

³¹⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/3285>

³¹⁰¹ <https://github.com/sphinx-doc/sphinx/issues/3363>

³¹⁰² <https://github.com/sphinx-doc/sphinx/issues/3214>

³¹⁰³ <https://github.com/sphinx-doc/sphinx/issues/3195>

³¹⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/3198>

³¹⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/3211>

³¹⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/3212>

³¹⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/3207>

³¹⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/3205>

³¹⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/3220>

³¹¹⁰ <https://github.com/sphinx-doc/sphinx/issues/3200>

³¹¹¹ <https://github.com/sphinx-doc/sphinx/issues/3228>

³¹¹² <https://github.com/sphinx-doc/sphinx/issues/2469>

- [#3183](#)³¹¹³: Randomized jump box order in generated index page.

14.123 Release 1.5 (released Dec 5, 2016)

Modifiche incompatibili

1.5a1

- latex, package fancybox is not any longer a dependency of sphinx.sty
- Use 'locales' as a default value of locale_dirs
- latex, package ifthen is not any longer a dependency of sphinx.sty
- latex, style file does not modify fancyvrb's Verbatim (also available as OriginalVerbatim) but uses sphinxVerbatim for name of custom wrapper.
- latex, package newfloat is not used (and not included) anymore (ref [#2660](#)³¹¹⁴; it was used since 1.3.4 and shipped with Sphinx since 1.4).
- latex, literal blocks in tables do not use OriginalVerbatim but sphinxVerbatimintable which handles captions and wraps lines (ref [#2704](#)³¹¹⁵).
- latex, replace pt by TeX equivalent bp if found in width or height attribute of an image.
- latex, if width or height attribute of an image is given with no unit, use px rather than ignore it.
- latex: Separate stylesheets of pygments to independent .sty file
- [#2454](#)³¹¹⁶: The filename of sourcelink is now changed. The value of html_sourcelink_suffix will be appended to the original filename (like index.rst.txt).
- sphinx.util.copy_static_entry() is now deprecated. Use sphinx.util.fileutil.copy_asset() instead.
- sphinx.util.osutil.filecopy() skips copying if the file has not been changed (ref: [#2510](#)³¹¹⁷, [#2753](#)³¹¹⁸)
- Internet Explorer 6-8, Opera 12.1x or Safari 5.1+ support is dropped because jQuery version is updated from 1.11.0 to 3.1.0 (ref: [#2634](#)³¹¹⁹, [#2773](#)³¹²⁰)
- QtHelpBuilder doesn't generate search page (ref: [#2352](#)³¹²¹)

³¹¹³ <https://github.com/sphinx-doc/sphinx/issues/3183>

³¹¹⁴ <https://github.com/sphinx-doc/sphinx/issues/2660>

³¹¹⁵ <https://github.com/sphinx-doc/sphinx/issues/2704>

³¹¹⁶ <https://github.com/sphinx-doc/sphinx/issues/2454>

³¹¹⁷ <https://github.com/sphinx-doc/sphinx/issues/2510>

³¹¹⁸ <https://github.com/sphinx-doc/sphinx/issues/2753>

³¹¹⁹ <https://github.com/sphinx-doc/sphinx/issues/2634>

³¹²⁰ <https://github.com/sphinx-doc/sphinx/issues/2773>

³¹²¹ <https://github.com/sphinx-doc/sphinx/issues/2352>

- QtHelpBuilder uses nonav theme instead of default one to improve readability.
- latex: To provide good default settings to Japanese documents, Sphinx uses `jreport` and `jsbook` as docclass if `language` is `ja`.
- `sphinx-quickstart` now allows a project version is empty
- Fix `:download:` role on epub/qthelp builder. They ignore the role because they don't support it.
- `sphinx.ext.viewcode` doesn't work on epub building by default. `viewcode_enable_epub` option
- `sphinx.ext.viewcode` disabled on singlehtml builder.
- Use make-mode of `sphinx-quickstart` by default. To disable this, use `-M` option
- Fix `genindex.html`, Sphinx's document template, link address to itself to satisfy xhtml standard.
- Use epub3 builder by default. And the old epub builder is renamed to epub2.
- Fix epub and epub3 builders that contained links to `genindex` even if `epub_use_index = False`.
- `html_translator_class` is now deprecated. Use `set_translator()` API instead.
- Drop python 2.6 and 3.3 support
- Drop epub3 builder's `epub3_page_progression_direction` option (use `epub3_writing_mode`).
- #2877³¹²²: Rename `latex_elements['footer']` to `latex_elements['atendofbody']`

1.5a2

- #2983³¹²³: Rename `epub3_description` and `epub3_contributor` to `epub_description` and `epub_contributor`.
- Remove `themes/basic/defindex.html`; no longer used
- Sphinx does not ship anymore (but still uses) LaTeX style file `fncychap`
- #2435³¹²⁴: Slim down quickstarted `conf.py`
- The `sphinx.sty` latex package does not load itself "hyperref", as this is done later in the preamble of the latex output via 'hyperref' key.
- Sphinx does not ship anymore a custom modified LaTeX style file `tabulary`. The non-modified package is used.
- #3057³¹²⁵: By default, footnote marks in latex PDF output are not preceded by a space anymore, `\sphinxBeforeFootnote` allows user customization if needed.

³¹²² <https://github.com/sphinx-doc/sphinx/issues/2877>

³¹²³ <https://github.com/sphinx-doc/sphinx/issues/2983>

³¹²⁴ <https://github.com/sphinx-doc/sphinx/issues/2435>

³¹²⁵ <https://github.com/sphinx-doc/sphinx/issues/3057>

- LaTeX target requires that option `hyperfootnotes` of package `hyperref` be left unchanged to its default (i.e. `true`) (refs: [#3022](#)³¹²⁶)

1.5 final

- [#2986](#)³¹²⁷: `themes/basic/defindex.html` is now deprecated
- Emit warnings that will be deprecated in Sphinx 1.6 by default. Users can change the behavior by setting the environment variable `PYTHONWARNINGS`. Please refer *Deprecazione dei Warning*.
- [#2454](#)³¹²⁸: new JavaScript variable `SOURCELINK_SUFFIX` is added

Deprecato

These features are removed in Sphinx 1.6:

- LDML format support in `i18n` feature
- `sphinx.addnodes.termsep`
- Some functions and classes in `sphinx.util.pycompat`: `zip_longest`, `product`, `all`, `any`, `next`, `open`, `class_types`, `base_exception`, `relpath`, `StringIO`, `BytesIO`. Please use the standard library version instead;

If any deprecation warning like `RemovedInSphinxXXXWarning` are displayed, please refer *Deprecazione dei Warning*.

Funzionalità aggiunte

1.5a1

- [#2951](#)³¹²⁹: Add `--implicit-namespaces` PEP-0420 support to `apidoc`.
- Add `:caption:` option for `sphinx.ext.inheritance_diagram`.
- [#2471](#)³¹³⁰: Add config variable for default doctest flags.
- Convert linkcheck builder to requests for better encoding handling
- [#2463](#)³¹³¹, [#2516](#)³¹³²: Add keywords of "meta" directive to search index
- `:maxdepth:` option of `toctree` affects `secnumdepth` (ref: [#2547](#)³¹³³)
- [#2575](#)³¹³⁴: Now `sphinx.ext.graphviz` allows `:align:` option

³¹²⁶ <https://github.com/sphinx-doc/sphinx/issues/3022>

³¹²⁷ <https://github.com/sphinx-doc/sphinx/issues/2986>

³¹²⁸ <https://github.com/sphinx-doc/sphinx/issues/2454>

³¹²⁹ <https://github.com/sphinx-doc/sphinx/issues/2951>

³¹³⁰ <https://github.com/sphinx-doc/sphinx/issues/2471>

³¹³¹ <https://github.com/sphinx-doc/sphinx/issues/2463>

³¹³² <https://github.com/sphinx-doc/sphinx/issues/2516>

³¹³³ <https://github.com/sphinx-doc/sphinx/issues/2547>

³¹³⁴ <https://github.com/sphinx-doc/sphinx/issues/2575>

- Show warnings if unknown key is specified to `latex_elements`
- Show warnings if no domains match with `primary_domain` (ref: #2001³¹³⁵)
- C++, show warnings when the kind of role is misleading for the kind of target it refers to (e.g., using the `class` role for a function).
- latex, writer abstracts more of text styling into customizable macros, e.g. the `visit_emphasis` will output `\sphinxstyleemphasis` rather than `\emph` (which may be in use elsewhere or in an added LaTeX package). See list at end of `sphinx.sty` (ref: #2686³¹³⁶)
- latex, public names for environments and parameters used by note, warning, and other admonition types, allowing full customizability from the 'preamble' key or an input file (ref: feature request #2674³¹³⁷, #2685³¹³⁸)
- latex, better computes column widths of some tables (as a result, there will be slight changes as tables now correctly fill the line width; ref: #2708³¹³⁹)
- latex, `sphinxVerbatim` environment is more easily customizable (ref: #2704³¹⁴⁰). In addition to already existing `VerbatimColor` and `VerbatimBorderColor`:
 - two lengths `\sphinxverbatimsep` and `\sphinxverbatimborder`,
 - booleans `\ifsphinxverbatimwithframe` and `\ifsphinxverbatimwraplines`.
- latex, captions for literal blocks inside tables are handled, and long code lines wrapped to fit table cell (ref: #2704³¹⁴¹)
- #2597³¹⁴²: Show warning messages as darkred
- latex, allow image dimensions using px unit (default is 96px=1in)
- Show warnings if invalid dimension units found
- #2650³¹⁴³: Add `--pdb` option to `setup.py` command
- latex, make the use of `\small` for code listings customizable (ref #2721³¹⁴⁴)
- #2663³¹⁴⁵: Add `--warning-is-error` option to `setup.py` command
- Show warnings if deprecated latex options are used
- Add `sphinx.config.ENUM` to check the config values is in candidates
- math: Add hyperlink marker to each equations in HTML output

³¹³⁵ <https://github.com/sphinx-doc/sphinx/issues/2001>

³¹³⁶ <https://github.com/sphinx-doc/sphinx/issues/2686>

³¹³⁷ <https://github.com/sphinx-doc/sphinx/issues/2674>

³¹³⁸ <https://github.com/sphinx-doc/sphinx/issues/2685>

³¹³⁹ <https://github.com/sphinx-doc/sphinx/issues/2708>

³¹⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/2704>

³¹⁴¹ <https://github.com/sphinx-doc/sphinx/issues/2704>

³¹⁴² <https://github.com/sphinx-doc/sphinx/issues/2597>

³¹⁴³ <https://github.com/sphinx-doc/sphinx/issues/2650>

³¹⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/2721>

³¹⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/2663>

- Add new theme nonav that doesn't include any navigation links. This is for any help generator like qthelp.
- #2680³¹⁴⁶: `sphinx.ext.todo` now emits warnings if `todo_emit_warnings` enabled. Also, it emits an additional event named `todo-defined` to handle the TODO entries in 3rd party extensions.
- Python domain signature parser now uses the `xref` mixin for 'exceptions', allowing exception classes to be autolinked.
- #2513³¹⁴⁷: Add `latex_engine` to switch the LaTeX engine by `conf.py`
- #2682³¹⁴⁸: C++, basic support for attributes (C++11 style and GNU style). The new configuration variables '`cpp_id_attributes`' and '`cpp_paren_attributes`' can be used to introduce custom attributes.
- #1958³¹⁴⁹: C++, add configuration variable '`cpp_index_common_prefix`' for removing prefixes from the index text of C++ objects.
- C++, added concept directive. Thanks to mickk-on-cpp.
- C++, added support for template introduction syntax. Thanks to mickk-on-cpp.
- #2725³¹⁵⁰: latex builder: allow to use user-defined template file (experimental)
- apidoc now avoids invalidating cached files by not writing to files whose content doesn't change. This can lead to significant performance wins if apidoc is run frequently.
- #2851³¹⁵¹: `sphinx.ext.math` emits missing-reference event if equation not found
- #1210³¹⁵²: `eqref` role now supports cross reference
- #2892³¹⁵³: Added `-a` (`--append-syspath`) option to `sphinx-apidoc`
- #1604³¹⁵⁴: epub3 builder: Obey font-related CSS when viewing in iBooks.
- #646³¹⁵⁵: `option` directive support `'` character as a part of options
- Add document about kindlegen and fix document structure for it.
- #2474³¹⁵⁶: Add `intersphinx_timeout` option to `sphinx.ext.intersphinx`
- #2926³¹⁵⁷: EPUB3 builder supports vertical mode (`epub3_writing_mode` option)
- #2695³¹⁵⁸: `build_sphinx` subcommand for `setuptools` handles exceptions as same as `sphinx-build` does

³¹⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/2680>

³¹⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/2513>

³¹⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/2682>

³¹⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/1958>

³¹⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/2725>

³¹⁵¹ <https://github.com/sphinx-doc/sphinx/issues/2851>

³¹⁵² <https://github.com/sphinx-doc/sphinx/issues/1210>

³¹⁵³ <https://github.com/sphinx-doc/sphinx/issues/2892>

³¹⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/1604>

³¹⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/646>

³¹⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/2474>

³¹⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/2926>

³¹⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/2695>

- [#326](#)³¹⁵⁹: `numref` role can also refer sections
- [#2916](#)³¹⁶⁰: `numref` role can also refer caption as an its linktext

1.5a2

- [#3008](#)³¹⁶¹: `linkcheck` builder ignores self-signed certificate URL
- [#3020](#)³¹⁶²: new `'geometry'` key to `latex_elements` whose default uses LaTeX style file `geometry.sty` to set page layout
- [#2843](#)³¹⁶³: Add `:start-at:` and `:end-at:` options to `literalinclude` directive
- [#2527](#)³¹⁶⁴: Add `:reversed:` option to `toctree` directive
- Add `-t` and `-d` option to `sphinx-quickstart` to support templating generated Sphinx project.
- [#3028](#)³¹⁶⁵: Add `{path}` and `{basename}` to the format of `figure_language_filename`
- new `'hyperref'` key in the `latex_elements` dictionary (ref [#3030](#)³¹⁶⁶)
- [#3022](#)³¹⁶⁷: Allow code-blocks in footnotes for LaTeX PDF output

1.5b1

- [#2513](#)³¹⁶⁸: A better default settings for XeLaTeX
- [#3096](#)³¹⁶⁹: `'maxlistdepth'` key to work around LaTeX list limitations
- [#3060](#)³¹⁷⁰: `autodoc` supports documentation for attributes of Enum class. Now `autodoc` render just the value of Enum attributes instead of Enum attribute representation.
- Add `--extensions` to `sphinx-quickstart` to support enable arbitrary extensions from command line (ref: [#2904](#)³¹⁷¹)
- [#3104](#)³¹⁷², [#3122](#)³¹⁷³: `'sphinxsetup'` for key=value styling of Sphinx LaTeX
- [#3071](#)³¹⁷⁴: `Autodoc`: Allow mocked module decorators to pass-through functions unchanged

³¹⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/326>

³¹⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/2916>

³¹⁶¹ <https://github.com/sphinx-doc/sphinx/issues/3008>

³¹⁶² <https://github.com/sphinx-doc/sphinx/issues/3020>

³¹⁶³ <https://github.com/sphinx-doc/sphinx/issues/2843>

³¹⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/2527>

³¹⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/3028>

³¹⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/3030>

³¹⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/3022>

³¹⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/2513>

³¹⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/3096>

³¹⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/3060>

³¹⁷¹ <https://github.com/sphinx-doc/sphinx/issues/2904>

³¹⁷² <https://github.com/sphinx-doc/sphinx/issues/3104>

³¹⁷³ <https://github.com/sphinx-doc/sphinx/issues/3122>

³¹⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/3071>

- #2495³¹⁷⁵: linkcheck: Allow skipping anchor checking using `linkcheck_anchors_ignore`
- #3083³¹⁷⁶: let Unicode no-break space act like LaTeX ~ (fixed #3019³¹⁷⁷)
- #3116³¹⁷⁸: allow word wrap in PDF output for inline literals (ref #3110³¹⁷⁹)
- #930³¹⁸⁰: sphinx-apidoc allow wildcards for excluding paths. Thanks to Nick Coghlan.
- #3121³¹⁸¹: add `inlineliteralwraps` option to control if inline literal word-wraps in latex

1.5 final

- #3095³¹⁸²: Add `tls_verify` and `tls_cacerts` to support self-signed HTTPS servers in linkcheck and intersphinx
- #2215³¹⁸³: `make.bat` generated by sphinx-quickstart can be called from another dir. Thanks to Timotheus Kampik.
- #3185³¹⁸⁴: Add new warning type `misc.highlighting_failure`

Bugs fixed

1.5a1

- #2707³¹⁸⁵: (latex) the column width is badly computed for tabular
- #2799³¹⁸⁶: Sphinx installs roles and directives automatically on importing sphinx module. Now Sphinx installs them on running application.
- `sphinx.ext.autodoc` crashes if target code imports `*` from mock modules by `autodoc_mock_imports`.
- #1953³¹⁸⁷: `Sphinx.add_node` does not add handlers the translator installed by `html_translator_class`
- #1797³¹⁸⁸: text builder inserts blank line on top
- #2894³¹⁸⁹: quickstart `main()` doesn't use `argv` argument

³¹⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/2495>

³¹⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/3083>

³¹⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/3019>

³¹⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/3116>

³¹⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/3110>

³¹⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/930>

³¹⁸¹ <https://github.com/sphinx-doc/sphinx/issues/3121>

³¹⁸² <https://github.com/sphinx-doc/sphinx/issues/3095>

³¹⁸³ <https://github.com/sphinx-doc/sphinx/issues/2215>

³¹⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/3185>

³¹⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/2707>

³¹⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/2799>

³¹⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/1953>

³¹⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/1797>

³¹⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/2894>

- [#2874³¹⁹⁰](#): gettext builder could not extract all text under the only directives
- [#2485³¹⁹¹](#): autosummary crashes with multiple source_suffix values
- [#1734³¹⁹²](#): Could not translate the caption of toctree directive
- Could not translate the content of meta directive (ref: [#1734³¹⁹³](#))
- [#2550³¹⁹⁴](#): external links are opened in help viewer
- [#2687³¹⁹⁵](#): Running Sphinx multiple times produces 'already registered' warnings

1.5a2

- [#2810³¹⁹⁶](#): Problems with pdflatex in an Italian document
- Use `latex_elements.papersize` to specify papersize of LaTeX in Makefile
- [#2988³¹⁹⁷](#): linkcheck: retry with GET request if denied HEAD request
- [#2990³¹⁹⁸](#): linkcheck raises "Can't convert 'bytes' object to str implicitly" error if linkcheck_anchors enabled
- [#3004³¹⁹⁹](#): Invalid link types "top" and "up" are used
- [#3009³²⁰⁰](#): Bad rendering of parsed-literals in LaTeX since Sphinx 1.4.4
- [#3000³²⁰¹](#): option directive generates invalid HTML anchors
- [#2984³²⁰²](#): Invalid HTML has been generated if `html_split_index` enabled
- [#2986³²⁰³](#): themes/basic/defindex.html should be changed for html5 friendly
- [#2987³²⁰⁴](#): Invalid HTML has been generated if multiple IDs are assigned to a list
- [#2891³²⁰⁵](#): HTML search does not provide all the results
- [#1986³²⁰⁶](#): Title in PDF Output
- [#147³²⁰⁷](#): Problem with latex chapter style
- [#3018³²⁰⁸](#): LaTeX problem with page layout dimensions and chapter titles

³¹⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/2874>

³¹⁹¹ <https://github.com/sphinx-doc/sphinx/issues/2485>

³¹⁹² <https://github.com/sphinx-doc/sphinx/issues/1734>

³¹⁹³ <https://github.com/sphinx-doc/sphinx/issues/1734>

³¹⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/2550>

³¹⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/2687>

³¹⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/2810>

³¹⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/2988>

³¹⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/2990>

³¹⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/3004>

³²⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/3009>

³²⁰¹ <https://github.com/sphinx-doc/sphinx/issues/3000>

³²⁰² <https://github.com/sphinx-doc/sphinx/issues/2984>

³²⁰³ <https://github.com/sphinx-doc/sphinx/issues/2986>

³²⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/2987>

³²⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/2891>

³²⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/1986>

³²⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/147>

³²⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/3018>

- Fix an issue with `\pysigline` in LaTeX style file (ref #3023³²⁰⁹)
- #3038³²¹⁰: `sphinx.ext.math*` raises `TypeError` if labels are duplicated
- #3031³²¹¹: incompatibility with LaTeX package `tocloft`
- #3003³²¹²: literal blocks in footnotes are not supported by Latex
- #3047³²¹³: spacing before footnote in pdf output is not coherent and allows breaks
- #3045³²¹⁴: HTML search index creator should ignore "raw" content if now html
- #3039³²¹⁵: English stemmer returns wrong word if the word is capitalized
- Fix make-mode Makefile template (ref #3056³²¹⁶, #2936³²¹⁷)

1.5b1

- #2432³²¹⁸: Fix unwanted `*` between varargs and keyword only args. Thanks to Alex Grönholm.
- #3062³²¹⁹: Failed to build PDF using 1.5a2 (undefined `\hypersetup` for Japanese documents since PR#3030³²²⁰)
- Better rendering of multiline signatures in html.
- #777³²²¹: LaTeX output "too deeply nested" (ref #3096³²²²)
- Let LaTeX image inclusion obey `scale` before `textwidth` fit (ref #2865³²²³, #3059³²²⁴)
- #3019³²²⁵: LaTeX fails on description of C function with arguments (ref #3083³²²⁶)
- fix latex inline literals where `< >` - gobbled a space

1.5 final

- #3069³²²⁷: Even if `'babel'` key is set to empty string, LaTeX output contains one `\addto\captions...`
- #3123³²²⁸: user `'babel'` key setting is not obeyed anymore

³²⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/3023>

³²¹⁰ <https://github.com/sphinx-doc/sphinx/issues/3038>

³²¹¹ <https://github.com/sphinx-doc/sphinx/issues/3031>

³²¹² <https://github.com/sphinx-doc/sphinx/issues/3003>

³²¹³ <https://github.com/sphinx-doc/sphinx/issues/3047>

³²¹⁴ <https://github.com/sphinx-doc/sphinx/issues/3045>

³²¹⁵ <https://github.com/sphinx-doc/sphinx/issues/3039>

³²¹⁶ <https://github.com/sphinx-doc/sphinx/issues/3056>

³²¹⁷ <https://github.com/sphinx-doc/sphinx/issues/2936>

³²¹⁸ <https://github.com/sphinx-doc/sphinx/issues/2432>

³²¹⁹ <https://github.com/sphinx-doc/sphinx/issues/3062>

³²²⁰ <https://github.com/sphinx-doc/sphinx/issues/3030>

³²²¹ <https://github.com/sphinx-doc/sphinx/issues/777>

³²²² <https://github.com/sphinx-doc/sphinx/issues/3096>

³²²³ <https://github.com/sphinx-doc/sphinx/issues/2865>

³²²⁴ <https://github.com/sphinx-doc/sphinx/issues/3059>

³²²⁵ <https://github.com/sphinx-doc/sphinx/issues/3019>

³²²⁶ <https://github.com/sphinx-doc/sphinx/issues/3083>

³²²⁷ <https://github.com/sphinx-doc/sphinx/issues/3069>

³²²⁸ <https://github.com/sphinx-doc/sphinx/issues/3123>

- [#3155](#)³²²⁹: Fix JavaScript for `html_sourcelink_suffix` fails with IE and Opera
- [#3085](#)³²³⁰: keep current directory after breaking build documentation. Thanks to Timotheus Kampik.
- [#3181](#)³²³¹: pLaTeX crashes with a section contains endash
- [#3180](#)³²³²: latex: add stretch/shrink between successive singleline or multipleline cpp signatures (ref [#3072](#)³²³³)
- [#3128](#)³²³⁴: globing images does not support .svgz file
- [#3015](#)³²³⁵: fix a broken test on Windows.
- [#1843](#)³²³⁶: Fix documentation of descriptor classes that have a custom metaclass. Thanks to Erik Bray.
- [#3190](#)³²³⁷: `util.split_docinfo` fails to parse multi-line field bodies
- [#3024](#)³²³⁸, [#3037](#)³²³⁹: In Python3, `application.Sphinx._log` crushed when the log message cannot be encoded into console encoding.

Testing

- To simplify, Sphinx uses external mock package even if `unittest.mock` exists.

14.124 Release 1.4.9 (released Nov 23, 2016)

Bugs fixed

- [#2936](#)³²⁴⁰: Fix doc/Makefile that can't build man because doc/man exists
- [#3058](#)³²⁴¹: Using the same 'caption' attribute in multiple 'toctree' directives results in warning / error
- [#3068](#)³²⁴²: Allow the '=' character in the -D option of sphinx-build.py
- [#3074](#)³²⁴³: `add_source_parser()` crashes in debug mode

³²²⁹ <https://github.com/sphinx-doc/sphinx/issues/3155>

³²³⁰ <https://github.com/sphinx-doc/sphinx/issues/3085>

³²³¹ <https://github.com/sphinx-doc/sphinx/issues/3181>

³²³² <https://github.com/sphinx-doc/sphinx/issues/3180>

³²³³ <https://github.com/sphinx-doc/sphinx/issues/3072>

³²³⁴ <https://github.com/sphinx-doc/sphinx/issues/3128>

³²³⁵ <https://github.com/sphinx-doc/sphinx/issues/3015>

³²³⁶ <https://github.com/sphinx-doc/sphinx/issues/1843>

³²³⁷ <https://github.com/sphinx-doc/sphinx/issues/3190>

³²³⁸ <https://github.com/sphinx-doc/sphinx/issues/3024>

³²³⁹ <https://github.com/sphinx-doc/sphinx/issues/3037>

³²⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/2936>

³²⁴¹ <https://github.com/sphinx-doc/sphinx/issues/3058>

³²⁴² <https://github.com/sphinx-doc/sphinx/issues/3068>

³²⁴³ <https://github.com/sphinx-doc/sphinx/issues/3074>

- [#3135](#)³²⁴⁴: `sphinx.ext.autodoc` crashes with plain Callable
- [#3150](#)³²⁴⁵: Fix query word splitter in JavaScript. It behaves as same as Python's regular expression.
- [#3093](#)³²⁴⁶: gettext build broken on substituted images.
- [#3093](#)³²⁴⁷: gettext build broken on image node under `note` directive.
- `imgmath`: crashes on showing error messages if image generation failed
- [#3117](#)³²⁴⁸: LaTeX writer crashes if admonition is placed before first section title
- [#3164](#)³²⁴⁹: Change search order of `sphinx.ext.inheritance_diagram`

14.125 Release 1.4.8 (released Oct 1, 2016)

Bugs fixed

- [#2996](#)³²⁵⁰: The wheel package of Sphinx got crash with `ImportError`

14.126 Release 1.4.7 (released Oct 1, 2016)

Bugs fixed

- [#2890](#)³²⁵¹: Quickstart should return an error consistently on all error conditions
- [#2870](#)³²⁵²: flatten genindex columns' heights.
- [#2856](#)³²⁵³: Search on generated HTML site doesn't find some symbols
- [#2882](#)³²⁵⁴: Fall back to a GET request on 403 status in linkcheck
- [#2902](#)³²⁵⁵: `jsdump.loads` fails to load search index if keywords starts with underscore
- [#2900](#)³²⁵⁶: Fix epub content.opf: add auto generated orphan files to spine.

³²⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/3135>

³²⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/3150>

³²⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/3093>

³²⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/3093>

³²⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/3117>

³²⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/3164>

³²⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/2996>

³²⁵¹ <https://github.com/sphinx-doc/sphinx/issues/2890>

³²⁵² <https://github.com/sphinx-doc/sphinx/issues/2870>

³²⁵³ <https://github.com/sphinx-doc/sphinx/issues/2856>

³²⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/2882>

³²⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/2902>

³²⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/2900>

- [#2899³²⁵⁷](#): Fix `hasdoc()` function in Jinja2 template. It will detect `genindex`, search also.
- [#2901³²⁵⁸](#): Fix epub result: skip creating links from image tags to original image files.
- [#2917³²⁵⁹](#): inline code is hyphenated on HTML
- [#1462³²⁶⁰](#): autosummary warns for namedtuple with attribute with trailing underscore
- Could not reference equations if `:nowrap:` option specified
- [#2873³²⁶¹](#): code-block overflow in latex (due to commas)
- [#1060³²⁶²](#), [#2056³²⁶³](#): `sphinx.ext.intersphinx`: broken links are generated if relative paths are used in `intersphinx_mapping`
- [#2931³²⁶⁴](#): code-block directive with same `:caption:` causes warning of duplicate target. Now `code-block` and `literalinclude` does not define hyperlink target using its caption automatically.
- [#2962³²⁶⁵](#): latex: missing label of `longtable`
- [#2968³²⁶⁶](#): autodoc: `show-inheritance` option breaks docstrings

14.127 Release 1.4.6 (released Aug 20, 2016)

Modifiche incompatibili

- [#2867³²⁶⁷](#): linkcheck builder crashes with six-1.4. Now Sphinx depends on six-1.5 or later

Bugs fixed

- `applehelp`: Sphinx crashes if `hiutil` or `codesign` commands not found
- Fix `make clean` abort issue when build dir contains regular files like `DS_Store`.
- Reduce epubcheck warnings/errors:
 - Fix DOCTYPE to `html5`
 - Change extension from `.html` to `.xhtml`.

³²⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/2899>

³²⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/2901>

³²⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/2917>

³²⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/1462>

³²⁶¹ <https://github.com/sphinx-doc/sphinx/issues/2873>

³²⁶² <https://github.com/sphinx-doc/sphinx/issues/1060>

³²⁶³ <https://github.com/sphinx-doc/sphinx/issues/2056>

³²⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/2931>

³²⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/2962>

³²⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/2968>

³²⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/2867>

- Disable search page on epub results
- [#2778](#)³²⁶⁸: Fix autodoc crashes if `obj.__dict__` is a property method and raises exception
- Fix duplicated toc in epub3 output.
- [#2775](#)³²⁶⁹: Fix failing linkcheck with servers not supporting identity encoding
- [#2833](#)³²⁷⁰: Fix formatting instance annotations in `ext.autodoc`.
- [#1911](#)³²⁷¹: `-D` option of `sphinx-build` does not override the `extensions` variable
- [#2789](#)³²⁷²: `sphinx.ext.intersphinx` generates wrong hyperlinks if the inventory is given
- parsing errors for caption of code-blocks are displayed in document (ref: [#2845](#)³²⁷³)
- [#2846](#)³²⁷⁴: `singlehtml` builder does not include figure numbers
- [#2816](#)³²⁷⁵: Fix data from builds cluttering the `Domain.initial_data` class attributes

14.128 Release 1.4.5 (released Jul 13, 2016)

Modifiche incompatibili

- latex, inclusion of non-inline images from image directive resulted in non-coherent whitespaces depending on original image width; new behaviour by necessity differs from earlier one in some cases. (ref: [#2672](#)³²⁷⁶)
- latex, use of `\includegraphics` to refer to Sphinx custom variant is deprecated; in future it will revert to original LaTeX macro, custom one already has alternative name `\sphinxincludegraphics`.

Funzionalità aggiunte

- new config option `latex_keep_old_macro_names`, defaults to `True`. If `False`, lets macros (for text styling) be defined only with `\sphinx`-prefixed names
- latex writer allows user customization of "shadowed" boxes (topics), via three length variables.
- woff-format web font files now supported by the epub builder.

³²⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/2778>

³²⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/2775>

³²⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/2833>

³²⁷¹ <https://github.com/sphinx-doc/sphinx/issues/1911>

³²⁷² <https://github.com/sphinx-doc/sphinx/issues/2789>

³²⁷³ <https://github.com/sphinx-doc/sphinx/issues/2845>

³²⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/2846>

³²⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/2816>

³²⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/2672>

Bugs fixed

- jsdump fix for python 3: fixes the HTML search on python > 3
- #2676³²⁷⁷: (latex) Error with verbatim text in captions since Sphinx 1.4.4
- #2629³²⁷⁸: memoir class crashes LaTeX. Fixed by `latex_keep_old_macro_names=False` (ref 2675)
- #2684³²⁷⁹: `sphinx.ext.intersphinx` crashes with six-1.4.1
- #2679³²⁸⁰: float package needed for 'figure_align': 'H' latex option
- #2671³²⁸¹: image directive may lead to inconsistent spacing in pdf
- #2705³²⁸²: `toctree` generates empty `bullet_list` if `:titlesonly:` specified
- #2479³²⁸³: `sphinx.ext.viewcode` uses python2 highlighter by default
- #2700³²⁸⁴: `HtmlHelp` builder has hard coded `index.html`
- latex, since 1.4.4 inline literal text is followed by spurious space
- #2722³²⁸⁵: C++, fix id generation for var/member declarations to include namespaces.
- latex, images (from image directive) in lists or quoted blocks did not obey indentation (fixed together with #2671³²⁸⁶)
- #2733³²⁸⁷: since Sphinx 1.4.4 `make latexpdf` generates lots of hyperref warnings
- #2731³²⁸⁸: `sphinx.ext.autodoc` does not access `property` methods which raises any exceptions
- #2666³²⁸⁹: C++, properly look up nested names involving constructors.
- #2579³²⁹⁰: Could not refer a label including both spaces and colons via `sphinx.ext.intersphinx`
- #2718³²⁹¹: Sphinx crashes if the document file is not readable
- #2699³²⁹²: hyperlinks in help HTMLs are broken if `html_file_suffix` is set

³²⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/2676>

³²⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/2629>

³²⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/2684>

³²⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/2679>

³²⁸¹ <https://github.com/sphinx-doc/sphinx/issues/2671>

³²⁸² <https://github.com/sphinx-doc/sphinx/issues/2705>

³²⁸³ <https://github.com/sphinx-doc/sphinx/issues/2479>

³²⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/2700>

³²⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/2722>

³²⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/2671>

³²⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/2733>

³²⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/2731>

³²⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/2666>

³²⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/2579>

³²⁹¹ <https://github.com/sphinx-doc/sphinx/issues/2718>

³²⁹² <https://github.com/sphinx-doc/sphinx/issues/2699>

- [#2723³²⁹³](#): extra spaces in latex pdf output from multirow cell
- [#2735³²⁹⁴](#): latexpdf Underfull \hbox (badness 10000) warnings from title page
- [#2667³²⁹⁵](#): latex crashes if resized images appeared in section title
- [#2763³²⁹⁶](#): (html) Provide default value for required alt attribute for image tags of SVG source, required to validate and now consistent w/ other formats.

14.129 Release 1.4.4 (released Jun 12, 2016)

Bugs fixed

- [#2630³²⁹⁷](#): latex: sphinx.sty notice environment formatting problem
- [#2632³²⁹⁸](#): Warning directives fail in quote environment latex build
- [#2633³²⁹⁹](#): Sphinx crashes with old styled indices
- Fix a `\begin{\minipage}` typo in sphinx.sty from 1.4.2 (ref: 68becb1)
- [#2622³³⁰⁰](#): Latex produces empty pages after title and table of contents
- [#2640³³⁰¹](#): 1.4.2 LaTeX crashes if code-block inside warning directive
- Let LaTeX use straight quotes also in inline code (ref [#2627³³⁰²](#))
- [#2351³³⁰³](#): latex crashes if enumerated lists are placed on footnotes
- [#2646³³⁰⁴](#): latex crashes if math contains twice empty lines
- [#2480³³⁰⁵](#): `sphinx.ext.autodoc`: memory addresses were shown
- latex: allow code-blocks appearing inside lists and quotes at maximal nesting depth (ref [#777³³⁰⁶](#), [#2624³³⁰⁷](#), [#2651³³⁰⁸](#))
- [#2635³³⁰⁹](#): Latex code directives produce inconsistent frames based on viewing resolution

³²⁹³ <https://github.com/sphinx-doc/sphinx/issues/2723>

³²⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/2735>

³²⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/2667>

³²⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/2763>

³²⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/2630>

³²⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/2632>

³²⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/2633>

³³⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/2622>

³³⁰¹ <https://github.com/sphinx-doc/sphinx/issues/2640>

³³⁰² <https://github.com/sphinx-doc/sphinx/issues/2627>

³³⁰³ <https://github.com/sphinx-doc/sphinx/issues/2351>

³³⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/2646>

³³⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/2480>

³³⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/777>

³³⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/2624>

³³⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/2651>

³³⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/2635>

- [#2639](#)³³¹⁰: Sphinx now bundles iftex.sty
- Failed to build PDF with framed.sty 0.95
- Sphinx now bundles needspace.sty

14.130 Release 1.4.3 (released Jun 5, 2016)

Bugs fixed

- [#2530](#)³³¹¹: got "Counter too large" error on building PDF if large numbered footnotes existed in admonitions
- width option of figure directive does not work if align option specified at same time (ref: [#2595](#)³³¹²)
- [#2590](#)³³¹³: The inputenc package breaks compiling under lualatex and xelatex
- [#2540](#)³³¹⁴: date on latex front page use different font
- Suppress "document isn't included in any toctree" warning if the document is included (ref: [#2603](#)³³¹⁵)
- [#2614](#)³³¹⁶: Some tables in PDF output will end up shifted if user sets non zero parindent in preamble
- [#2602](#)³³¹⁷: URL redirection breaks the hyperlinks generated by sphinx.ext.intersphinx
- [#2613](#)³³¹⁸: Show warnings if merged extensions are loaded
- [#2619](#)³³¹⁹: make sure amstext LaTeX package always loaded (ref: d657225, 488ee52, 9d82cad and [#2615](#)³³²⁰)
- [#2593](#)³³²¹: latex crashes if any figures in the table

³³¹⁰ <https://github.com/sphinx-doc/sphinx/issues/2639>

³³¹¹ <https://github.com/sphinx-doc/sphinx/issues/2530>

³³¹² <https://github.com/sphinx-doc/sphinx/issues/2595>

³³¹³ <https://github.com/sphinx-doc/sphinx/issues/2590>

³³¹⁴ <https://github.com/sphinx-doc/sphinx/issues/2540>

³³¹⁵ <https://github.com/sphinx-doc/sphinx/issues/2603>

³³¹⁶ <https://github.com/sphinx-doc/sphinx/issues/2614>

³³¹⁷ <https://github.com/sphinx-doc/sphinx/issues/2602>

³³¹⁸ <https://github.com/sphinx-doc/sphinx/issues/2613>

³³¹⁹ <https://github.com/sphinx-doc/sphinx/issues/2619>

³³²⁰ <https://github.com/sphinx-doc/sphinx/issues/2615>

³³²¹ <https://github.com/sphinx-doc/sphinx/issues/2593>

14.131 Release 1.4.2 (released May 29, 2016)

Funzionalità aggiunte

- Now `suppress_warnings` accepts following configurations (ref: #2451³³²², #2466³³²³):
 - `app.add_node`
 - `app.add_directive`
 - `app.add_role`
 - `app.add_generic_role`
 - `app.add_source_parser`
 - `image.data_uri`
 - `image.nonlocal_uri`
- #2453³³²⁴: LaTeX writer allows page breaks in topic contents; and their horizontal extent now fits in the line width (with shadow in margin). Also warning-type admonitions allow page breaks and their vertical spacing has been made more coherent with the one for hint-type notices (ref #2446³³²⁵).
- #2459³³²⁶: the framing of literal code-blocks in LaTeX output (and not only the code lines themselves) obey the indentation in lists or quoted blocks.
- #2343³³²⁷: the long source lines in code-blocks are wrapped (without modifying the line numbering) in LaTeX output (ref #1534³³²⁸, #2304³³²⁹).

Bugs fixed

- #2370³³³⁰: the equations are slightly misaligned in LaTeX writer
- #1817³³³¹, #2077³³³²: suppress pep8 warnings on `conf.py` generated by `sphinx-quickstart`
- #2407³³³³: building docs crash if document includes large data image URIs
- #2436³³³⁴: Sphinx does not check version by `needs_sphinx` if loading extensions failed

³³²² <https://github.com/sphinx-doc/sphinx/issues/2451>

³³²³ <https://github.com/sphinx-doc/sphinx/issues/2466>

³³²⁴ <https://github.com/sphinx-doc/sphinx/issues/2453>

³³²⁵ <https://github.com/sphinx-doc/sphinx/issues/2446>

³³²⁶ <https://github.com/sphinx-doc/sphinx/issues/2459>

³³²⁷ <https://github.com/sphinx-doc/sphinx/issues/2343>

³³²⁸ <https://github.com/sphinx-doc/sphinx/issues/1534>

³³²⁹ <https://github.com/sphinx-doc/sphinx/issues/2304>

³³³⁰ <https://github.com/sphinx-doc/sphinx/issues/2370>

³³³¹ <https://github.com/sphinx-doc/sphinx/issues/1817>

³³³² <https://github.com/sphinx-doc/sphinx/issues/2077>

³³³³ <https://github.com/sphinx-doc/sphinx/issues/2407>

³³³⁴ <https://github.com/sphinx-doc/sphinx/issues/2436>

- [#2397](#)³³³⁵: Setup shorthandoff for Turkish documents
- [#2447](#)³³³⁶: VerbatimBorderColor wrongly used also for captions of PDF
- [#2456](#)³³³⁷: C++, fix crash related to document merging (e.g., singlehtml and Latex builders).
- [#2446](#)³³³⁸: latex(pdf) sets local tables of contents (or more generally topic nodes) in unbreakable boxes, causes overflow at bottom
- [#2476](#)³³³⁹: Omit MathJax markers if :nowrap: is given
- [#2465](#)³³⁴⁰: latex builder fails in case no caption option is provided to toctree directive
- Sphinx crashes if self referenced toctree found
- [#2481](#)³³⁴¹: spelling mistake for mecab search splitter. Thanks to Naoki Sato.
- [#2309](#)³³⁴²: Fix could not refer "indirect hyperlink targets" by ref-role
- intersphinx fails if mapping URL contains any port
- [#2088](#)³³⁴³: intersphinx crashes if the mapping URL requires basic auth
- [#2304](#)³³⁴⁴: auto line breaks in latexpdf codeblocks
- [#1534](#)³³⁴⁵: Word wrap long lines in Latex verbatim blocks
- [#2460](#)³³⁴⁶: too much white space on top of captioned literal blocks in PDF output
- Show error reason when multiple math extensions are loaded (ref: [#2499](#)³³⁴⁷)
- [#2483](#)³³⁴⁸: any figure number was not assigned if figure title contains only non text objects
- [#2501](#)³³⁴⁹: Unicode subscript numbers are normalized in LaTeX
- [#2492](#)³³⁵⁰: Figure directive with :figwidth: generates incorrect Latex-code
- The caption of figure is always put on center even if :align: was specified
- [#2526](#)³³⁵¹: LaTeX writer crashes if the section having only images

³³³⁵ <https://github.com/sphinx-doc/sphinx/issues/2397>

³³³⁶ <https://github.com/sphinx-doc/sphinx/issues/2447>

³³³⁷ <https://github.com/sphinx-doc/sphinx/issues/2456>

³³³⁸ <https://github.com/sphinx-doc/sphinx/issues/2446>

³³³⁹ <https://github.com/sphinx-doc/sphinx/issues/2476>

³³⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/2465>

³³⁴¹ <https://github.com/sphinx-doc/sphinx/issues/2481>

³³⁴² <https://github.com/sphinx-doc/sphinx/issues/2309>

³³⁴³ <https://github.com/sphinx-doc/sphinx/issues/2088>

³³⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/2304>

³³⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/1534>

³³⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/2460>

³³⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/2499>

³³⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/2483>

³³⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/2501>

³³⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/2492>

³³⁵¹ <https://github.com/sphinx-doc/sphinx/issues/2526>

- [#2522](#)³³⁵²: Sphinx touches mo files under installed directory that caused permission error.
- [#2536](#)³³⁵³: C++, fix crash when an immediately nested scope has the same name as the current scope.
- [#2555](#)³³⁵⁴: Fix crash on any-references with unicode.
- [#2517](#)³³⁵⁵: wrong bookmark encoding in PDF if using LuaLaTeX
- [#2521](#)³³⁵⁶: generated Makefile causes BSD make crashed if sphinx-build not found
- [#2470](#)³³⁵⁷: typing backport package causes autodoc errors with python 2.7
- `sphinx.ext.intersphinx` crashes if non-string value is used for key of `intersphinx_mapping`
- [#2518](#)³³⁵⁸: `intersphinx_mapping` disallows non alphanumeric keys
- [#2558](#)³³⁵⁹: unpack error on devhelp builder
- [#2561](#)³³⁶⁰: Info builder crashes when a footnote contains a link
- [#2565](#)³³⁶¹: The descriptions of objects generated by `sphinx.ext.autosummary` overflow lines at LaTeX writer
- Extend `pdflatex` config in `sphinx.sty` to subparagraphs (ref: [#2551](#)³³⁶²)
- [#2445](#)³³⁶³: `rst_prolog` and `rst_epilog` affect to non reST sources
- [#2576](#)³³⁶⁴: `sphinx.ext.imgmath` crashes if subprocess raises error
- [#2577](#)³³⁶⁵: `sphinx.ext.imgmath`: Invalid argument are passed to `dvisvgm`
- [#2556](#)³³⁶⁶: Xapian search does not work with Python 3
- [#2581](#)³³⁶⁷: The search doesn't work if `language="es"` (Spanish)
- [#2382](#)³³⁶⁸: Adjust spacing after abbreviations on figure numbers in LaTeX writer
- [#2383](#)³³⁶⁹: The generated footnote by `latex_show_urls` overflows lines

³³⁵² <https://github.com/sphinx-doc/sphinx/issues/2522>

³³⁵³ <https://github.com/sphinx-doc/sphinx/issues/2536>

³³⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/2555>

³³⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/2517>

³³⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/2521>

³³⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/2470>

³³⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/2518>

³³⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/2558>

³³⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/2561>

³³⁶¹ <https://github.com/sphinx-doc/sphinx/issues/2565>

³³⁶² <https://github.com/sphinx-doc/sphinx/issues/2551>

³³⁶³ <https://github.com/sphinx-doc/sphinx/issues/2445>

³³⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/2576>

³³⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/2577>

³³⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/2556>

³³⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/2581>

³³⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/2382>

³³⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/2383>

- #2497³³⁷⁰, #2552³³⁷¹: The label of search button does not fit for the button itself

14.132 Release 1.4.1 (released Apr 12, 2016)

Modifiche incompatibili

- The default format of `today_fmt` and `html_last_updated_fmt` is back to `strftime` format again. Locale Date Markup Language is also supported for backward compatibility until Sphinx 1.5.

Traduzioni

- Added Welsh translation, thanks to Geraint Palmer.
- Added Greek translation, thanks to Stelios Vitalis.
- Added Esperanto translation, thanks to Dinu Gherman.
- Added Hindi translation, thanks to Purnank H. Ghumalia.
- Added Romanian translation, thanks to Razvan Stefanescu.

Bugs fixed

- C++, added support for `extern` and `thread_local`.
- C++, type declarations are now using the prefixes `typedef`, `using`, and `type`, depending on the style of declaration.
- #2413³³⁷²: C++, fix crash on duplicate declarations
- #2394³³⁷³: Sphinx crashes when `html_last_updated_fmt` is invalid
- #2408³³⁷⁴: dummy builder not available in `Makefile` and `make.bat`
- #2412³³⁷⁵: hyperlink targets are broken in LaTeX builder
- `figure` directive crashes if non paragraph item is given as caption
- #2418³³⁷⁶: time formats no longer allowed in `today_fmt`
- #2395³³⁷⁷: Sphinx crashes if unicode character in image filename

³³⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/2497>

³³⁷¹ <https://github.com/sphinx-doc/sphinx/issues/2552>

³³⁷² <https://github.com/sphinx-doc/sphinx/issues/2413>

³³⁷³ <https://github.com/sphinx-doc/sphinx/issues/2394>

³³⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/2408>

³³⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/2412>

³³⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/2418>

³³⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/2395>

- [#2396](#)³³⁷⁸: "too many values to unpack" in genindex-single
- [#2405](#)³³⁷⁹: numref link in PDF jumps to the wrong location
- [#2414](#)³³⁸⁰: missing number in PDF hyperlinks to code listings
- [#2440](#)³³⁸¹: wrong import for gmtime. Thanks to Uwe L. Korn.

14.133 Release 1.4 (released Mar 28, 2016)

Modifiche incompatibili

- Drop PorterStemmer package support. Use PyStemmer instead of PorterStemmer to accelerate stemming.
- `sphinx_rtd_theme` has become optional. Please install it manually. Refs [#2087](#)³³⁸², [#2086](#)³³⁸³, [#1845](#)³³⁸⁴ and [#2097](#)³³⁸⁵. Thanks to Victor Zverovich.
- [#2231](#)³³⁸⁶: Use `DUrole` instead of `DUspan` for custom roles in LaTeX writer. It enables to take title of roles as an argument of custom macros.
- [#2022](#)³³⁸⁷: 'Thumbs.db' and '.DS_Store' are added to `exclude_patterns` default values in `conf.py` that will be provided on `sphinx-quickstart`.
- [#2027](#)³³⁸⁸, [#2208](#)³³⁸⁹: The `html_title` accepts string values only. And the `None` value cannot be accepted.
- `sphinx.ext.graphviz`: show graph image in inline by default
- [#2060](#)³³⁹⁰, [#2224](#)³³⁹¹: The `manpage` role now generate `sphinx.addnodes.manpage` node instead of `sphinx.addnodes.literal_emphasis` node.
- [#2022](#)³³⁹²: `html_extra_path` also copies dotfiles in the extra directory, and refers to `exclude_patterns` to exclude extra files and directories.
- [#2300](#)³³⁹³: enhance `autoclass::` to use the docstring of `__new__` if `__init__` method's is missing of empty

³³⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/2396>

³³⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/2405>

³³⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/2414>

³³⁸¹ <https://github.com/sphinx-doc/sphinx/issues/2440>

³³⁸² <https://github.com/sphinx-doc/sphinx/issues/2087>

³³⁸³ <https://github.com/sphinx-doc/sphinx/issues/2086>

³³⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/1845>

³³⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/2097>

³³⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/2231>

³³⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/2022>

³³⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/2027>

³³⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/2208>

³³⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/2060>

³³⁹¹ <https://github.com/sphinx-doc/sphinx/issues/2224>

³³⁹² <https://github.com/sphinx-doc/sphinx/issues/2022>

³³⁹³ <https://github.com/sphinx-doc/sphinx/issues/2300>

- [#2251](#)³³⁹⁴: Previously, under glossary directives, multiple terms for one definition are converted into single `term` node and the each terms in the term node are separated by `termsep` node. In new implementation, each terms are converted into individual `term` nodes and `termsep` node is removed. By this change, output layout of every builders are changed a bit.
- The default highlight language is now Python 3. This means that source code is highlighted as Python 3 (which is mostly a superset of Python 2), and no parsing is attempted to distinguish valid code. To get the old behavior back, add `highlight_language = "python"` to `conf.py`.
- **Locale Date Markup Language**³³⁹⁵ like `"MMMM dd, YYYY"` is default format for `today_fmt` and `html_last_updated_fmt`. However `strftime` format like `"%B %d, %Y"` is also supported for backward compatibility until Sphinx 1.5. Later format will be disabled from Sphinx 1.5.
- [#2327](#)³³⁹⁶: `latex_use_parts` is deprecated now. Use `latex_toplevel_sectioning` instead.
- [#2337](#)³³⁹⁷: Use `\url{URL}` macro instead of `\href{URL}{URL}` in LaTeX writer.
- [#1498](#)³³⁹⁸: manpage writer: don't make whole of item in definition list bold if it includes strong node.
- [#582](#)³³⁹⁹: Remove hint message from quick search box for html output.
- [#2378](#)³⁴⁰⁰: Sphinx now bundles `newfloat.sty`

Funzionalità aggiunte

- [#2092](#)³⁴⁰¹: add `todo` directive support in `napoleon` package.
- [#1962](#)³⁴⁰²: when adding directives, roles or nodes from an extension, warn if such an element is already present (built-in or added by another extension).
- [#1909](#)³⁴⁰³: Add `"doc"` references to Intersphinx inventories.
- C++ type alias support (e.g., `... type:: T = int`).
- C++ template support for classes, functions, type aliases, and variables ([#1729](#)³⁴⁰⁴, [#1314](#)³⁴⁰⁵).

³³⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/2251>

³³⁹⁵ https://unicode.org/reports/tr35/tr35-dates.html#Date_Format_Patterns

³³⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/2327>

³³⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/2337>

³³⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/1498>

³³⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/582>

³⁴⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/2378>

³⁴⁰¹ <https://github.com/sphinx-doc/sphinx/issues/2092>

³⁴⁰² <https://github.com/sphinx-doc/sphinx/issues/1962>

³⁴⁰³ <https://github.com/sphinx-doc/sphinx/issues/1909>

³⁴⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/1729>

³⁴⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/1314>

- C++, added new scope management directives `namespace-push` and `namespace-pop`.
- #1970³⁴⁰⁶: Keyboard shortcuts to navigate Next and Previous topics
- Intersphinx: Added support for fetching Intersphinx inventories with URLs using HTTP basic auth.
- C++, added support for template parameter in function info field lists.
- C++, added support for pointers to member (function).
- #2113³⁴⁰⁷: Allow `:class:` option to code-block directive.
- #2192³⁴⁰⁸: `Imgmath` (`pngmath` with `svg` support).
- #2200³⁴⁰⁹: Support XeTeX and LuaTeX for the LaTeX builder.
- #1906³⁴¹⁰: Use `xcolor` over `color` for `colorbox` where available for LaTeX output.
- #2216³⁴¹¹: Texinputs makefile improvements.
- #2170³⁴¹²: Support for Chinese language search index.
- #2214³⁴¹³: Add `sphinx.ext.githubpages` to publish the docs on GitHub Pages
- #1030³⁴¹⁴: Make page reference names for `latex_show_pagerefs` translatable
- #2162³⁴¹⁵: Add `Sphinx.add_source_parser()` to add `source_suffix` and `source_parsers` from extension
- #2207³⁴¹⁶: Add `sphinx.parsers.Parser` class; a base class for new parsers
- #656³⁴¹⁷: Add `graphviz_dot` option to `graphviz` directives to switch the `dot` command
- #1939³⁴¹⁸: Added the dummy builder: syntax check without output.
- #2230³⁴¹⁹: Add `math_number_all` option to number all displayed math in math extensions
- #2235³⁴²⁰: `needs_sphinx` supports micro version comparison
- #2282³⁴²¹: Add "language" attribute to html tag in the "basic" theme

³⁴⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/1970>

³⁴⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/2113>

³⁴⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/2192>

³⁴⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/2200>

³⁴¹⁰ <https://github.com/sphinx-doc/sphinx/issues/1906>

³⁴¹¹ <https://github.com/sphinx-doc/sphinx/issues/2216>

³⁴¹² <https://github.com/sphinx-doc/sphinx/issues/2170>

³⁴¹³ <https://github.com/sphinx-doc/sphinx/issues/2214>

³⁴¹⁴ <https://github.com/sphinx-doc/sphinx/issues/1030>

³⁴¹⁵ <https://github.com/sphinx-doc/sphinx/issues/2162>

³⁴¹⁶ <https://github.com/sphinx-doc/sphinx/issues/2207>

³⁴¹⁷ <https://github.com/sphinx-doc/sphinx/issues/656>

³⁴¹⁸ <https://github.com/sphinx-doc/sphinx/issues/1939>

³⁴¹⁹ <https://github.com/sphinx-doc/sphinx/issues/2230>

³⁴²⁰ <https://github.com/sphinx-doc/sphinx/issues/2235>

³⁴²¹ <https://github.com/sphinx-doc/sphinx/issues/2282>

- [#1779](#)³⁴²²: Add EPUB 3 builder
- [#1751](#)³⁴²³: Add `todo_link_only` to avoid file path and line indication on `todolist`. Thanks to Francesco Montesano.
- [#2199](#)³⁴²⁴: Use `imagesize` package to obtain size of images.
- [#1099](#)³⁴²⁵: Add configurable retries to the linkcheck builder. Thanks to Alex Gaynor. Also don't check anchors starting with !.
- [#2300](#)³⁴²⁶: enhance `autoclass::` to use the docstring of `__new__` if `__init__` method's is missing of empty
- [#1858](#)³⁴²⁷: Add `Sphinx.add_enumerable_node()` to add enumerable nodes for numfig feature
- [#1286](#)³⁴²⁸, [#2099](#)³⁴²⁹: Add `sphinx.ext.autosectionlabel` extension to allow reference sections using its title. Thanks to Tadhg O'Higgins.
- [#1854](#)³⁴³⁰: Allow to choose Janome for Japanese splitter.
- [#1853](#)³⁴³¹: support custom text splitter on html search with `language='ja'`.
- [#2320](#)³⁴³²: classifier of glossary terms can be used for index entries grouping key The classifier also be used for translation. See also *Glossario*.
- [#2308](#)³⁴³³: Define `\tablecontinued` macro to redefine the style of continued label for longtables.
- Select an image by similarity if multiple images are globbed by `.. image:: filename.*`
- [#1921](#)³⁴³⁴: Support figure substitutions by `language` and `figure_language_filename`
- [#2245](#)³⁴³⁵: Add `latex_elements["passoptionstopackages"]` option to call `PassOptionsToPackages` in early stage of preambles.
- [#2340](#)³⁴³⁶: Math extension: support alignment of multiple equations for MathJax.
- [#2338](#)³⁴³⁷: Define `\titleref` macro to redefine the style of title-reference roles.

³⁴²² <https://github.com/sphinx-doc/sphinx/issues/1779>

³⁴²³ <https://github.com/sphinx-doc/sphinx/issues/1751>

³⁴²⁴ <https://github.com/sphinx-doc/sphinx/issues/2199>

³⁴²⁵ <https://github.com/sphinx-doc/sphinx/issues/1099>

³⁴²⁶ <https://github.com/sphinx-doc/sphinx/issues/2300>

³⁴²⁷ <https://github.com/sphinx-doc/sphinx/issues/1858>

³⁴²⁸ <https://github.com/sphinx-doc/sphinx/issues/1286>

³⁴²⁹ <https://github.com/sphinx-doc/sphinx/issues/2099>

³⁴³⁰ <https://github.com/sphinx-doc/sphinx/issues/1854>

³⁴³¹ <https://github.com/sphinx-doc/sphinx/issues/1853>

³⁴³² <https://github.com/sphinx-doc/sphinx/issues/2320>

³⁴³³ <https://github.com/sphinx-doc/sphinx/issues/2308>

³⁴³⁴ <https://github.com/sphinx-doc/sphinx/issues/1921>

³⁴³⁵ <https://github.com/sphinx-doc/sphinx/issues/2245>

³⁴³⁶ <https://github.com/sphinx-doc/sphinx/issues/2340>

³⁴³⁷ <https://github.com/sphinx-doc/sphinx/issues/2338>

- Define `\menuselection` and `\accelerator` macros to redefine the style of `menuselection` roles.
- Define `\crossref` macro to redefine the style of references
- #2301³⁴³⁸: Texts in the classic html theme should be hyphenated.
- #2355³⁴³⁹: Define `\termref` macro to redefine the style of `term` roles.
- Add `suppress_warnings` to suppress arbitrary warning message (experimental)
- #2229³⁴⁴⁰: Fix no warning is given for unknown options
- #2327³⁴⁴¹: Add `latex_toplevel_sectioning` to switch the top level sectioning of LaTeX document.

Bugs fixed

- #1913³⁴⁴²: C++, fix assert bug for enumerators in next-to-global and global scope.
- C++, fix parsing of 'signed char' and 'unsigned char' as types.
- C++, add missing support for 'friend' functions.
- C++, add missing support for virtual base classes (thanks to Rapptz).
- C++, add support for final classes.
- C++, fix parsing of types prefixed with 'enum'.
- #2023³⁴⁴³: Dutch search support uses Danish stemming info.
- C++, add support for user-defined literals.
- #1804³⁴⁴⁴: Now html output wraps overflowed long-line-text in the sidebar. Thanks to Hassen ben tanfous.
- #2183³⁴⁴⁵: Fix porterstemmer causes `make json` to fail.
- #1899³⁴⁴⁶: Ensure list is sent to `OptParse`.
- #2164³⁴⁴⁷: Fix wrong check for `pdftex` inside `sphinx.sty` (for `graphicx` package option).
- #2165³⁴⁴⁸, #2218³⁴⁴⁹: Remove faulty and non-need conditional from `sphinx.sty`.
- Fix broken LaTeX code is generated if unknown language is given

³⁴³⁸ <https://github.com/sphinx-doc/sphinx/issues/2301>

³⁴³⁹ <https://github.com/sphinx-doc/sphinx/issues/2355>

³⁴⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/2229>

³⁴⁴¹ <https://github.com/sphinx-doc/sphinx/issues/2327>

³⁴⁴² <https://github.com/sphinx-doc/sphinx/issues/1913>

³⁴⁴³ <https://github.com/sphinx-doc/sphinx/issues/2023>

³⁴⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/1804>

³⁴⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/2183>

³⁴⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/1899>

³⁴⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/2164>

³⁴⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/2165>

³⁴⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/2218>

- [#1944](#)³⁴⁵⁰: Fix rst_prolog breaks file-wide metadata
- [#2074](#)³⁴⁵¹: make gettext should use canonical relative paths for .pot. Thanks to anatoly techtonik.
- [#2311](#)³⁴⁵²: Fix sphinx.ext.inheritance_diagram raises AttributeError
- [#2251](#)³⁴⁵³: Line breaks in .rst files are transferred to .pot files in a wrong way.
- [#794](#)³⁴⁵⁴: Fix date formatting in latex output is not localized
- Remove image/gif from supported_image_types of LaTeX writer ([#2272](#)³⁴⁵⁵)
- Fix ValueError is raised if LANGUAGE is empty string
- Fix unpack warning is shown when the directives generated from Sphinx.add_crossref_type is used
- The default highlight language is now default. This means that source code is highlighted as Python 3 (which is mostly a superset of Python 2) if possible. To get the old behavior back, add highlight_language = "python" to conf.py.
- [#2329](#)³⁴⁵⁶: Refresh environment forcedly if source directory has changed.
- [#2331](#)³⁴⁵⁷: Fix code-blocks are filled by block in dvi; remove xcdraw option from xcolor package
- Fix the confval type checker emits warnings if unicode is given to confvals which expects string value
- [#2360](#)³⁴⁵⁸: Fix numref in LaTeX output is broken
- [#2361](#)³⁴⁵⁹: Fix additional paragraphs inside the "compound" directive are indented
- [#2364](#)³⁴⁶⁰: Fix KeyError 'rootSymbol' on Sphinx upgrade from older version.
- [#2348](#)³⁴⁶¹: Move amsmath and amssymb to before fontpkg on LaTeX writer.
- [#2368](#)³⁴⁶²: Ignore emacs lock files like .#foo.rst by default.
- [#2262](#)³⁴⁶³: literal_block and its caption has been separated by pagebreak in LaTeX output.
- [#2319](#)³⁴⁶⁴: Fix table counter is overridden by code-block's in LaTeX. Thanks to jfbu.

³⁴⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/1944>

³⁴⁵¹ <https://github.com/sphinx-doc/sphinx/issues/2074>

³⁴⁵² <https://github.com/sphinx-doc/sphinx/issues/2311>

³⁴⁵³ <https://github.com/sphinx-doc/sphinx/issues/2251>

³⁴⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/794>

³⁴⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/2272>

³⁴⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/2329>

³⁴⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/2331>

³⁴⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/2360>

³⁴⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/2361>

³⁴⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/2364>

³⁴⁶¹ <https://github.com/sphinx-doc/sphinx/issues/2348>

³⁴⁶² <https://github.com/sphinx-doc/sphinx/issues/2368>

³⁴⁶³ <https://github.com/sphinx-doc/sphinx/issues/2262>

³⁴⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/2319>

- Fix unpack warning if combined with 3rd party domain extensions.
- #1153³⁴⁶⁵: Fix figures in sidebar causes latex build error.
- #2358³⁴⁶⁶: Fix user-preamble could not override the tocdepth definition.
- #2358³⁴⁶⁷: Reduce tocdepth if `part` or `chapter` is used for `top_sectionlevel`
- #2351³⁴⁶⁸: Fix footnote spacing
- #2363³⁴⁶⁹: Fix `toctree()` in templates generates broken links in `SingleHTMLBuilder`.
- #2366³⁴⁷⁰: Fix empty hyperref is generated on `toctree` in `HTML` builder.

Documentation

- #1757³⁴⁷¹: Fix for usage of `html_last_updated_fmt`. Thanks to Ralf Hemmecke.

14.134 Release 1.3.6 (released Feb 29, 2016)

Funzionalità aggiunte

- #1873³⁴⁷², #1876³⁴⁷³, #2278³⁴⁷⁴: Add `page_source_suffix` html context variable. This should be introduced with `source_parsers` feature. Thanks for Eric Holscher.

Bugs fixed

- #2265³⁴⁷⁵: Fix `babel` is used in spite of disabling it on `latex_elements`
- #2295³⁴⁷⁶: Avoid mutating dictionary errors while enumerating members in `autodoc` with `Python 3`
- #2291³⁴⁷⁷: Fix `pdflatex` "Counter too large" error from footnotes inside tables of contents
- #2292³⁴⁷⁸: Fix some footnotes disappear from `LaTeX` output

³⁴⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/1153>

³⁴⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/2358>

³⁴⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/2358>

³⁴⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/2351>

³⁴⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/2363>

³⁴⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/2366>

³⁴⁷¹ <https://github.com/sphinx-doc/sphinx/issues/1757>

³⁴⁷² <https://github.com/sphinx-doc/sphinx/issues/1873>

³⁴⁷³ <https://github.com/sphinx-doc/sphinx/issues/1876>

³⁴⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/2278>

³⁴⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/2265>

³⁴⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/2295>

³⁴⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/2291>

³⁴⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/2292>

- [#2287](#)³⁴⁷⁹: `sphinx.transforms.Locale` always uses rst parser. Sphinx i18n feature should support parsers that specified `source_parsers`.
- [#2290](#)³⁴⁸⁰: Fix `sphinx.ext.mathbase` use of `amsfonts` may break user choice of math fonts
- [#2324](#)³⁴⁸¹: Print a hint how to increase the recursion limit when it is hit.
- [#1565](#)³⁴⁸², [#2229](#)³⁴⁸³: Revert new warning; the new warning will be triggered from version 1.4 on.
- [#2329](#)³⁴⁸⁴: Refresh environment forcedly if source directory has changed.
- [#2019](#)³⁴⁸⁵: Fix the domain objects in search result are not escaped

14.135 Release 1.3.5 (released Jan 24, 2016)

Bugs fixed

- Fix line numbers was not shown on warnings in LaTeX and texinfo builders
- Fix filenames were not shown on warnings of citations
- Fix line numbers was not shown on warnings in LaTeX and texinfo builders
- Fix line numbers was not shown on warnings of indices
- [#2026](#)³⁴⁸⁶: Fix LaTeX builder raises error if parsed-literal includes links
- [#2243](#)³⁴⁸⁷: Ignore strange docstring types for classes, do not crash
- [#2247](#)³⁴⁸⁸: Fix [#2205](#)³⁴⁸⁹ breaks make html for definition list with classifiers that contains regular-expression like string
- [#1565](#)³⁴⁹⁰: Sphinx will now emit a warning that highlighting was skipped if the syntax is incorrect for `code-block`, `literalinclude` and so on.
- [#2211](#)³⁴⁹¹: Fix paragraphs in table cell doesn't work in Latex output
- [#2253](#)³⁴⁹²: `:pyobject:` option of `literalinclude` directive can't detect indented body block when the block starts with blank or comment lines.

³⁴⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/2287>

³⁴⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/2290>

³⁴⁸¹ <https://github.com/sphinx-doc/sphinx/issues/2324>

³⁴⁸² <https://github.com/sphinx-doc/sphinx/issues/1565>

³⁴⁸³ <https://github.com/sphinx-doc/sphinx/issues/2229>

³⁴⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/2329>

³⁴⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/2019>

³⁴⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/2026>

³⁴⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/2243>

³⁴⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/2247>

³⁴⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/2205>

³⁴⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/1565>

³⁴⁹¹ <https://github.com/sphinx-doc/sphinx/issues/2211>

³⁴⁹² <https://github.com/sphinx-doc/sphinx/issues/2253>

- Fix TOC is not shown when no `:maxdepth:` for toctrees (ref: #771³⁴⁹³)
- Fix warning message for `:numref:` if target is in orphaned doc (ref: #2244³⁴⁹⁴)

14.136 Release 1.3.4 (released Jan 12, 2016)

Bugs fixed

- #2134³⁴⁹⁵: Fix figure caption with reference causes latex build error
- #2094³⁴⁹⁶: Fix rubric with reference not working in Latex
- #2147³⁴⁹⁷: Fix literalinclude code in latex does not break in pages
- #1833³⁴⁹⁸: Fix email addresses is showed again if latex_show_urls is not None
- #2176³⁴⁹⁹: sphinx.ext.graphviz: use <object> instead of to embed svg
- #967³⁵⁰⁰: Fix SVG inheritance diagram is not hyperlinked (clickable)
- #1237³⁵⁰¹: Fix footnotes not working in definition list in LaTeX
- #2168³⁵⁰²: Fix raw directive does not work for text writer
- #2171³⁵⁰³: Fix cannot linkcheck url with unicode
- #2182³⁵⁰⁴: LaTeX: support image file names with more than 1 dots
- #2189³⁵⁰⁵: Fix previous sibling link for first file in subdirectory uses last file, not intended previous from root toctree
- #2003³⁵⁰⁶: Fix decode error under python2 (only) when make linkcheck is run
- #2186³⁵⁰⁷: Fix LaTeX output of mathbb in math
- #1480³⁵⁰⁸, #2188³⁵⁰⁹: LaTeX: Support math in section titles
- #2071³⁵¹⁰: Fix same footnote in more than two section titles => LaTeX/PDF Bug

³⁴⁹³ <https://github.com/sphinx-doc/sphinx/issues/771>

³⁴⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/2244>

³⁴⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/2134>

³⁴⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/2094>

³⁴⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/2147>

³⁴⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/1833>

³⁴⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/2176>

³⁵⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/967>

³⁵⁰¹ <https://github.com/sphinx-doc/sphinx/issues/1237>

³⁵⁰² <https://github.com/sphinx-doc/sphinx/issues/2168>

³⁵⁰³ <https://github.com/sphinx-doc/sphinx/issues/2171>

³⁵⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/2182>

³⁵⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/2189>

³⁵⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/2003>

³⁵⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/2186>

³⁵⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/1480>

³⁵⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/2188>

³⁵¹⁰ <https://github.com/sphinx-doc/sphinx/issues/2071>

- #2040³⁵¹¹: Fix UnicodeDecodeError in sphinx-apidoc when author contains non-ascii characters
- #2193³⁵¹²: Fix shutil.SameFileError if source directory and destination directory are same
- #2178³⁵¹³: Fix unparsable C++ cross-reference when referencing a function with :cpp:any:
- #2206³⁵¹⁴: Fix Sphinx latex doc build failed due to a footnotes
- #2201³⁵¹⁵: Fix wrong table caption for tables with over 30 rows
- #2213³⁵¹⁶: Set <blockquote> in the classic theme to fit with <p>
- #1815³⁵¹⁷: Fix linkcheck does not raise an exception if warniserror set to true and link is broken
- #2197³⁵¹⁸: Fix slightly cryptic error message for missing index.rst file
- #1894³⁵¹⁹: Unlisted phony targets in quickstart Makefile
- #2125³⁵²⁰: Fix unifies behavior of collapsed fields (GroupedField and TypedField)
- #1408³⁵²¹: Check latex_logo validity before copying
- #771³⁵²²: Fix latex output doesn't set tocdepth
- #1820³⁵²³: On Windows, console coloring is broken with colorama version 0.3.3. Now Sphinx use colorama>=0.3.5 to avoid this problem.
- #2072³⁵²⁴: Fix footnotes in chapter-titles do not appear in PDF output
- #1580³⁵²⁵: Fix paragraphs in longtable don't work in Latex output
- #1366³⁵²⁶: Fix centered image not centered in latex
- #1860³⁵²⁷: Fix man page using :samp: with braces - font doesn't reset
- #1610³⁵²⁸: Sphinx crashes in Japanese indexing in some systems
- Fix Sphinx crashes if mecab initialization failed

³⁵¹¹ <https://github.com/sphinx-doc/sphinx/issues/2040>

³⁵¹² <https://github.com/sphinx-doc/sphinx/issues/2193>

³⁵¹³ <https://github.com/sphinx-doc/sphinx/issues/2178>

³⁵¹⁴ <https://github.com/sphinx-doc/sphinx/issues/2206>

³⁵¹⁵ <https://github.com/sphinx-doc/sphinx/issues/2201>

³⁵¹⁶ <https://github.com/sphinx-doc/sphinx/issues/2213>

³⁵¹⁷ <https://github.com/sphinx-doc/sphinx/issues/1815>

³⁵¹⁸ <https://github.com/sphinx-doc/sphinx/issues/2197>

³⁵¹⁹ <https://github.com/sphinx-doc/sphinx/issues/1894>

³⁵²⁰ <https://github.com/sphinx-doc/sphinx/issues/2125>

³⁵²¹ <https://github.com/sphinx-doc/sphinx/issues/1408>

³⁵²² <https://github.com/sphinx-doc/sphinx/issues/771>

³⁵²³ <https://github.com/sphinx-doc/sphinx/issues/1820>

³⁵²⁴ <https://github.com/sphinx-doc/sphinx/issues/2072>

³⁵²⁵ <https://github.com/sphinx-doc/sphinx/issues/1580>

³⁵²⁶ <https://github.com/sphinx-doc/sphinx/issues/1366>

³⁵²⁷ <https://github.com/sphinx-doc/sphinx/issues/1860>

³⁵²⁸ <https://github.com/sphinx-doc/sphinx/issues/1610>

- #2160³⁵²⁹: Fix broken TOC of PDFs if section includes an image
- #2172³⁵³⁰: Fix dysfunctional admonition `\py@lightbox` in `sphinx.sty`. Thanks to jfbu.
- #2198³⁵³¹, #2205 <<https://github.com/sphinx-doc/sphinx/issues/2205>>`_: make `gettext` generate broken msgid for definition lists.
- #2062³⁵³²: Escape characters in doctests are treated incorrectly with Python 2.
- #2225³⁵³³: Fix if the option does not begin with dash, linking is not performed
- #2226³⁵³⁴: Fix math is not HTML-encoded when `:nowrap:` is given (`jsmath`, `mathjax`)
- #1601³⁵³⁵, #2220³⁵³⁶: 'any' role breaks extended domains behavior. Affected extensions doesn't support `resolve_any_xref` and `resolve_xref` returns problematic node instead of `None`. `sphinxcontrib-httpdomain` is one of them.
- #2229³⁵³⁷: Fix no warning is given for unknown options

14.137 Release 1.3.3 (released Dec 2, 2015)

Bugs fixed

- #2177³⁵³⁸: Fix parallel hangs
- #2012³⁵³⁹: Fix exception occurred if `numfig_format` is invalid
- #2142³⁵⁴⁰: Provide non-minified JS code in `sphinx/search/non-minified-js/*.js` for source distribution on PyPI.
- #2148³⁵⁴¹: Error while building devhelp target with non-ASCII document.

³⁵²⁹ <https://github.com/sphinx-doc/sphinx/issues/2160>

³⁵³⁰ <https://github.com/sphinx-doc/sphinx/issues/2172>

³⁵³¹ <https://github.com/sphinx-doc/sphinx/issues/2198>

³⁵³² <https://github.com/sphinx-doc/sphinx/issues/2062>

³⁵³³ <https://github.com/sphinx-doc/sphinx/issues/2225>

³⁵³⁴ <https://github.com/sphinx-doc/sphinx/issues/2226>

³⁵³⁵ <https://github.com/sphinx-doc/sphinx/issues/1601>

³⁵³⁶ <https://github.com/sphinx-doc/sphinx/issues/2220>

³⁵³⁷ <https://github.com/sphinx-doc/sphinx/issues/2229>

³⁵³⁸ <https://github.com/sphinx-doc/sphinx/issues/2177>

³⁵³⁹ <https://github.com/sphinx-doc/sphinx/issues/2012>

³⁵⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/2142>

³⁵⁴¹ <https://github.com/sphinx-doc/sphinx/issues/2148>

14.138 Release 1.3.2 (released Nov 29, 2015)

Funzionalità aggiunte

- [#1935](#)³⁵⁴²: Make "numfig_format" overridable in latex_elements.

Bugs fixed

- [#1976](#)³⁵⁴³: Avoid "2.0" version of Babel because it doesn't work with Windows environment.
- Add a "default.css" stylesheet (which imports "classic.css") for compatibility
- [#1788](#)³⁵⁴⁴: graphviz extension raises exception when caption option is present.
- [#1789](#)³⁵⁴⁵: :pyobject: option of literalinclude directive includes following lines after class definitions
- [#1790](#)³⁵⁴⁶: literalinclude strips empty lines at the head and tail
- [#1802](#)³⁵⁴⁷: load plugin themes automatically when theme.conf use it as 'inherit'. Thanks to Takayuki Hirai.
- [#1794](#)³⁵⁴⁸: custom theme extended from alabaster or sphinx_rtd_theme can't find base theme.
- [#1834](#)³⁵⁴⁹: compatibility for Docutils 0.13: handle_io_errors keyword argument for docutils.io.FileInput cause TypeError.
- [#1823](#)³⁵⁵⁰: '.' as <module_path> for sphinx-apidoc cause an unfriendly error. Now '.' is converted to absolute path automatically.
- Fix a crash when setting up extensions which do not support metadata.
- [#1784](#)³⁵⁵¹: Provide non-minified JS code in sphinx/search/non-minified-js/*.js
- [#1822](#)³⁵⁵², [#1892](#)³⁵⁵³: Fix regression for [#1061](#)³⁵⁵⁴. autosummary can't generate doc for imported members since Sphinx 1.3b3. Thanks to Eric Larson.

³⁵⁴² <https://github.com/sphinx-doc/sphinx/issues/1935>

³⁵⁴³ <https://github.com/sphinx-doc/sphinx/issues/1976>

³⁵⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/1788>

³⁵⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/1789>

³⁵⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/1790>

³⁵⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/1802>

³⁵⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/1794>

³⁵⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/1834>

³⁵⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/1823>

³⁵⁵¹ <https://github.com/sphinx-doc/sphinx/issues/1784>

³⁵⁵² <https://github.com/sphinx-doc/sphinx/issues/1822>

³⁵⁵³ <https://github.com/sphinx-doc/sphinx/issues/1892>

³⁵⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/1061>

- #1793³⁵⁵⁵, #1819³⁵⁵⁶: "see also" misses a linebreak in text output. Thanks to Takayuki Hirai.
- #1780³⁵⁵⁷, #1866³⁵⁵⁸: "make text" shows "class" keyword twice. Thanks to Takayuki Hirai.
- #1871³⁵⁵⁹: Fix for LaTeX output of tables with one column and multirows.
- Work around the lack of the `HTMLParserError` exception in Python 3.5.
- #1949³⁵⁶⁰: Use `safe_getattr` in the coverage builder to avoid aborting with descriptors that have custom behavior.
- #1915³⁵⁶¹: Do not generate smart quotes in doc field type annotations.
- #1796³⁵⁶²: On py3, automated .mo building caused `UnicodeDecodeError`.
- #1923³⁵⁶³: Use babel features only if the babel latex element is nonempty.
- #1942³⁵⁶⁴: Fix a `KeyError` in websupport.
- #1903³⁵⁶⁵: Fix strange id generation for glossary terms.
- `make text` will crush if a definition list item has more than 1 classifiers as: `term : classifier1 : classifier2`.
- #1855³⁵⁶⁶: `make gettext` generates broken po file for definition lists with classifier.
- #1869³⁵⁶⁷: Fix problems when dealing with files containing non-ASCII characters. Thanks to Marvin Schmidt.
- #1798³⁵⁶⁸: Fix building LaTeX with references in titles.
- #1725³⁵⁶⁹: On py2 environment, doctest with using non-ASCII characters causes 'ascii' codec can't decode byte exception.
- #1540³⁵⁷⁰: Fix `RuntimeError` with circular referenced toctree
- #1983³⁵⁷¹: i18n translation feature breaks references which uses section name.
- #1990³⁵⁷²: Use caption of toctree to title of tableofcontents in LaTeX

³⁵⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/1793>

³⁵⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/1819>

³⁵⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/1780>

³⁵⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/1866>

³⁵⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/1871>

³⁵⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/1949>

³⁵⁶¹ <https://github.com/sphinx-doc/sphinx/issues/1915>

³⁵⁶² <https://github.com/sphinx-doc/sphinx/issues/1796>

³⁵⁶³ <https://github.com/sphinx-doc/sphinx/issues/1923>

³⁵⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/1942>

³⁵⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/1903>

³⁵⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/1855>

³⁵⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/1869>

³⁵⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/1798>

³⁵⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/1725>

³⁵⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/1540>

³⁵⁷¹ <https://github.com/sphinx-doc/sphinx/issues/1983>

³⁵⁷² <https://github.com/sphinx-doc/sphinx/issues/1990>

- [#1987](#)³⁵⁷³: Fix ampersand is ignored in `:menuselection:` and `:guilabel:` on LaTeX builder
- [#1994](#)³⁵⁷⁴: More supporting non-standard parser (like recomcommonmark parser) for Translation and WebSupport feature. Now `node.rawsource` is fall backed to `node.astext()` during Docutils transforming.
- [#1989](#)³⁵⁷⁵: "make blahblah" on Windows indicate help messages for sphinx-build every time. It was caused by wrong `make.bat` that generated by Sphinx 1.3.0/1.3.1.
- On Py2 environment, `conf.py` that is generated by sphinx-quickstart should have `u` prefixed config value for 'version' and 'release'.
- [#2102](#)³⁵⁷⁶: On Windows + Py3, using `|today|` and non-ASCII date format will raise `UnicodeEncodeError`.
- [#1974](#)³⁵⁷⁷: `UnboundLocalError`: local variable 'domain' referenced before assignment when using any role and `sphinx.ext.intersphinx` in same time.
- [#2121](#)³⁵⁷⁸: multiple words search doesn't find pages when words across on the page title and the page content.
- [#1884](#)³⁵⁷⁹, [#1885](#)³⁵⁸⁰: plug-in html themes cannot inherit another plug-in theme. Thanks to Suzumizaki.
- [#1818](#)³⁵⁸¹: `sphinx.ext.todo` directive generates broken html class attribute as 'admonition-' when *language* is specified with non-ASCII linguistic area like 'ru' or 'ja'. To fix this, now `todo` directive can use `:class:` option.
- [#2140](#)³⁵⁸²: Fix footnotes in table has broken in LaTeX
- [#2127](#)³⁵⁸³: MecabBinder for html searching feature doesn't work with Python 3. Thanks to Tomoko Uchida.

³⁵⁷³ <https://github.com/sphinx-doc/sphinx/issues/1987>

³⁵⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/1994>

³⁵⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/1989>

³⁵⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/2102>

³⁵⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/1974>

³⁵⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/2121>

³⁵⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/1884>

³⁵⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/1885>

³⁵⁸¹ <https://github.com/sphinx-doc/sphinx/issues/1818>

³⁵⁸² <https://github.com/sphinx-doc/sphinx/issues/2140>

³⁵⁸³ <https://github.com/sphinx-doc/sphinx/issues/2127>

14.139 Release 1.3.1 (released Mar 17, 2015)

Bugs fixed

- [#1769](#)³⁵⁸⁴: allows generating quickstart files/dirs for destination dir that doesn't overwrite existent files/dirs. Thanks to WAKAYAMA shirou.
- [#1773](#)³⁵⁸⁵: sphinx-quickstart doesn't accept non-ASCII character as a option argument.
- [#1766](#)³⁵⁸⁶: the message "least Python 2.6 to run" is at best misleading.
- [#1772](#)³⁵⁸⁷: cross reference in docstrings like `:param .write:` breaks building.
- [#1770](#)³⁵⁸⁸, [#1774](#)³⁵⁸⁹: `literalinclude` with empty file occurs exception. Thanks to Takayuki Hirai.
- [#1777](#)³⁵⁹⁰: Sphinx 1.3 can't load extra theme. Thanks to tell-k.
- [#1776](#)³⁵⁹¹: `source_suffix = ['.rst']` cause unfriendly error on prior version.
- [#1771](#)³⁵⁹²: automated `.mo` building doesn't work properly.
- [#1783](#)³⁵⁹³: Autodoc: Python2 Allow unicode string in `__all__`. Thanks to Jens Hedegaard Nielsen.
- [#1781](#)³⁵⁹⁴: Setting `html_domain_indices` to a list raises a type check warnings.

14.140 Release 1.3 (released Mar 10, 2015)

Modifiche incompatibili

- Roles `ref`, `term` and `menuse1` now don't generate [emphasis](#)³⁵⁹⁵ nodes anymore. If you want to keep italic style, adapt your stylesheet.
- Role `numref` uses `%s` as special character to indicate position of figure numbers instead `#` symbol.

³⁵⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/1769>

³⁵⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/1773>

³⁵⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/1766>

³⁵⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/1772>

³⁵⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/1770>

³⁵⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/1774>

³⁵⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/1777>

³⁵⁹¹ <https://github.com/sphinx-doc/sphinx/issues/1776>

³⁵⁹² <https://github.com/sphinx-doc/sphinx/issues/1771>

³⁵⁹³ <https://github.com/sphinx-doc/sphinx/issues/1783>

³⁵⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/1781>

³⁵⁹⁵ <https://docutils.sourceforge.io/docs/ref/rst/roles.html#emphasis>

Funzionalità aggiunte

- Add convenience directives and roles to the C++ domain: directive `cpp:var` as alias for `cpp:member`, role `:cpp:var` as alias for `:cpp:member`, and role `any` for cross-reference to any C++ declaration. [#1577](#)³⁵⁹⁶, [#1744](#)³⁵⁹⁷
- The `source_suffix` config value can now be a list of multiple suffixes.
- Add the ability to specify source parsers by source suffix with the `source_parsers` config value.
- [#1675](#)³⁵⁹⁸: A new builder, `AppleHelpBuilder`, has been added that builds Apple Help Books.

Bugs fixed

- 1.3b3 change breaks a previous gettext output that contains duplicated msgid such as "foo bar" and "version changes in 1.3: foo bar".
- [#1745](#)³⁵⁹⁹: latex builder cause maximum recursion depth exceeded when a footnote has a footnote mark itself.
- [#1748](#)³⁶⁰⁰: `SyntaxError` in `sphinx/ext/ifconfig.py` with Python 2.6.
- [#1658](#)³⁶⁰¹, [#1750](#)³⁶⁰²: No link created (and warning given) if option does not begin with -, / or +. Thanks to Takayuki Hirai.
- [#1753](#)³⁶⁰³: C++, added missing support for more complex declarations.
- [#1700](#)³⁶⁰⁴: Add `:caption:` option for `toctree`.
- [#1742](#)³⁶⁰⁵: `:name:` option is provided for `toctree`, `code-block` and `literalinclude` directives.
- [#1756](#)³⁶⁰⁶: Incorrect section titles in search that was introduced from 1.3b3.
- [#1746](#)³⁶⁰⁷: C++, fixed name lookup procedure, and added missing lookups in declarations.
- [#1765](#)³⁶⁰⁸: C++, fix old id generation to use fully qualified names.

³⁵⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/1577>

³⁵⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/1744>

³⁵⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/1675>

³⁵⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/1745>

³⁶⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/1748>

³⁶⁰¹ <https://github.com/sphinx-doc/sphinx/issues/1658>

³⁶⁰² <https://github.com/sphinx-doc/sphinx/issues/1750>

³⁶⁰³ <https://github.com/sphinx-doc/sphinx/issues/1753>

³⁶⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/1700>

³⁶⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/1742>

³⁶⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/1756>

³⁶⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/1746>

³⁶⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/1765>

Documentation

- [#1651](#)³⁶⁰⁹: Add `vartype` field description for python domain.

14.141 Release 1.3b3 (released Feb 24, 2015)

Modifiche incompatibili

- Dependency requirement updates: Docutils 0.11, Pygments 2.0
- The `gettext_enables` config value has been renamed to `gettext_additional_targets`.
- [#1735](#)³⁶¹⁰: Use <https://docs.python.org/> instead of `http` protocol. It was used for `sphinx.ext.intersphinx` and some documentation.

Funzionalità aggiunte

- [#1346](#)³⁶¹¹: Add new default theme;
 - Add 'alabaster' theme.
 - Add 'sphinx_rtd_theme' theme.
 - The 'default' html theme has been renamed to 'classic'. 'default' is still available, however it will emit notice a recommendation that using new 'alabaster' theme.
- Added `highlight_options` configuration value.
- The `language` config value is now available in the HTML templates.
- The `env-updated` event can now return a value, which is interpreted as an iterable of additional docnames that need to be rewritten.
- [#772](#)³⁶¹²: Support for scoped and unscoped enums in C++. Enumerators in unscoped enums are injected into the parent scope in addition to the enum scope.
- Add `todo_include_todos` config option to quickstart conf file, handled as described in documentation.
- HTML breadcrumb items tag has class "nav-item" and "nav-item-N" (like nav-item-0, 1, 2...).
- New option `sphinx-quickstart --use-make-mode` for generating Makefile that use `sphinx-build make-mode`.

³⁶⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/1651>

³⁶¹⁰ <https://github.com/sphinx-doc/sphinx/issues/1735>

³⁶¹¹ <https://github.com/sphinx-doc/sphinx/issues/1346>

³⁶¹² <https://github.com/sphinx-doc/sphinx/issues/772>

- [#1235](#)³⁶¹³: `i18n`: several node can be translated if it is set to `gettext_additional_targets` in `conf.py`. Supported nodes are:
 - 'literal-block'
 - 'doctest-block'
 - 'raw'
 - 'image'
- [#1227](#)³⁶¹⁴: Add `html_scaled_image_link` config option to `conf.py`, to control scaled image link.

Bugs fixed

- LaTeX writer now generates correct markup for cells spanning multiple rows.
- [#1674](#)³⁶¹⁵: Do not crash if a module's `__all__` is not a list of strings.
- [#1629](#)³⁶¹⁶: Use `VerbatimBorderColor` to add frame to code-block in LaTeX
- On windows, `make-mode` didn't work on Win32 platform if Sphinx was invoked as `python sphinx-build.py`.
- [#1687](#)³⁶¹⁷: `linkcheck` now treats 401 Unauthorized responses as "working".
- [#1690](#)³⁶¹⁸: `toctrees` with `glob` option now can also contain entries for single documents with explicit title.
- [#1591](#)³⁶¹⁹: `html` search results for C++ elements now has correct interpage links.
- `bizstyle` theme: nested long title pages make long breadcrumb that breaks page layout.
- `bizstyle` theme: all breadcrumb items become 'Top' on some mobile browser (iPhone5s safari).
- [#1722](#)³⁶²⁰: restore `toctree()` template function behavior that was changed at 1.3b1.
- [#1732](#)³⁶²¹: `i18n`: localized table caption raises exception.
- [#1718](#)³⁶²²: `:numref:` does not work with capital letters in the label
- [#1630](#)³⁶²³: resolve CSS conflicts, `div.container` css target for literal block wrapper now renamed to `div.literal-block-wrapper`.

³⁶¹³ <https://github.com/sphinx-doc/sphinx/issues/1235>

³⁶¹⁴ <https://github.com/sphinx-doc/sphinx/issues/1227>

³⁶¹⁵ <https://github.com/sphinx-doc/sphinx/issues/1674>

³⁶¹⁶ <https://github.com/sphinx-doc/sphinx/issues/1629>

³⁶¹⁷ <https://github.com/sphinx-doc/sphinx/issues/1687>

³⁶¹⁸ <https://github.com/sphinx-doc/sphinx/issues/1690>

³⁶¹⁹ <https://github.com/sphinx-doc/sphinx/issues/1591>

³⁶²⁰ <https://github.com/sphinx-doc/sphinx/issues/1722>

³⁶²¹ <https://github.com/sphinx-doc/sphinx/issues/1732>

³⁶²² <https://github.com/sphinx-doc/sphinx/issues/1718>

³⁶²³ <https://github.com/sphinx-doc/sphinx/issues/1630>

- `sphinx.util.pycompat` has been restored in its backwards-compatibility; slated for removal in Sphinx 1.4.
- #1719³⁶²⁴: LaTeX writer does not respect `numref_format` option in captions

14.142 Release 1.3b2 (released Dec 5, 2014)

Modifiche incompatibili

- update bundled `ez_setup.py` for `setuptools-7.0` that requires Python 2.6 or later.

Funzionalità aggiunte

- #1597³⁶²⁵: Added possibility to return a new template name from `html-page-context`.
- PR#314³⁶²⁶, #1150³⁶²⁷: Configuration values are now checked for their type. A warning is raised if the configured and the default value do not have the same type and do not share a common non-trivial base class.

Bugs fixed

- PR#311³⁶²⁸: `sphinx-quickstart` does not work on python 3.4.
- Fix `autodoc_docstring_signature` not working with signatures in class docstrings.
- Rebuilding cause crash unexpectedly when source files were added.
- #1607³⁶²⁹: Fix a crash when building latexpdf with "howto" class
- #1251³⁶³⁰: Fix again. Sections which depth are lower than `:tocdepth:` should not be shown on `localtoc` sidebar.
- `make-mode` didn't work on Win32 platform if Sphinx was installed by wheel package.

³⁶²⁴ <https://github.com/sphinx-doc/sphinx/issues/1719>

³⁶²⁵ <https://github.com/sphinx-doc/sphinx/issues/1597>

³⁶²⁶ <https://github.com/sphinx-doc/sphinx/issues/314>

³⁶²⁷ <https://github.com/sphinx-doc/sphinx/issues/1150>

³⁶²⁸ <https://github.com/sphinx-doc/sphinx/issues/311>

³⁶²⁹ <https://github.com/sphinx-doc/sphinx/issues/1607>

³⁶³⁰ <https://github.com/sphinx-doc/sphinx/issues/1251>

14.143 Release 1.3b1 (released Oct 10, 2014)

Modifiche incompatibili

- Dropped support for Python 2.5, 3.1 and 3.2.
- Dropped support for Docutils versions up to 0.9.
- Removed the `sphinx.ext.oldcmakup` extension.
- The deprecated config values `exclude_trees`, `exclude_dirnames` and `unused_docs` have been removed.
- A new node, `sphinx.addnodes.literal_strong`, has been added, for text that should appear literally (i.e. no smart quotes) in strong font. Custom writers will have to be adapted to handle this node.
- [PR#269³⁶³¹](#), [#1476³⁶³²](#): replace `<tt>` tag by `<code>`. User customized stylesheets should be updated If the css contain some styles for `tt` tag. Thanks to Takeshi Komiya.
- [#1543³⁶³³](#): `templates_path` is automatically added to `exclude_patterns` to avoid reading autosummary rst templates in the templates directory.
- Custom domains should implement the new `Domain.resolve_any_xref` method to make the any role work properly.
- gettext builder: gettext doesn't emit uuid information to generated pot files by default. Please set `True` to `gettext_uuid` to emit uuid information. Additionally, if the `python-levenshtein` 3rd-party package is installed, it will improve the calculation time.
- gettext builder: disable extracting/apply 'index' node by default. Please set 'index' to `gettext_enables` to enable extracting index entries.
- [PR#307³⁶³⁴](#): Add frame to code-block in LaTeX. Thanks to Takeshi Komiya.

Funzionalità aggiunte

- Add support for Python 3.4.
- Add support for Docutils 0.12
- Added `sphinx.ext.napoleon` extension for NumPy and Google style docstring support.
- Added support for parallel reading (parsing) of source files with the `sphinx-build -j` option. Third-party extensions will need to be checked for compatibility and may need to be adapted if they store information in the build environment object. See `env-merge-info`.

³⁶³¹ <https://github.com/sphinx-doc/sphinx/issues/269>

³⁶³² <https://github.com/sphinx-doc/sphinx/issues/1476>

³⁶³³ <https://github.com/sphinx-doc/sphinx/issues/1543>

³⁶³⁴ <https://github.com/sphinx-doc/sphinx/issues/307>

- Added the `any` role that can be used to find a cross-reference of *any* type in *any* domain. Custom domains should implement the new `Domain.resolve_any_xref` method to make this work properly.
- Exception logs now contain the last 10 messages emitted by Sphinx.
- Added support for extension versions (a string returned by `setup()`, these can be shown in the traceback log files). Version requirements for extensions can be specified in projects using the new `needs_extensions` config value.
- Changing the default role within a document with the `default-role`³⁶³⁵ directive is now supported.
- [PR#214](#)³⁶³⁶: Added stemming support for 14 languages, so that the built-in document search can now handle these. Thanks to Shibukawa Yoshiki.
- [PR#296](#)³⁶³⁷, [PR#303](#)³⁶³⁸, [#76](#)³⁶³⁹: `numfig` feature: Assign numbers to figures, tables and code-blocks. This feature is configured with `numfig`, `numfig_secnum_depth` and `numfig_format`. Also `numref` role is available. Thanks to Takeshi Komiya.
- [PR#202](#)³⁶⁴⁰: Allow `"."` and `"~"` prefixed references in `:param:` doc fields for Python.
- [PR#184](#)³⁶⁴¹: Add `autodoc_mock_imports`, allowing to mock imports of external modules that need not be present when autodocumenting.
- [#925](#)³⁶⁴²: Allow list-typed config values to be provided on the command line, like `-D key=val1,val2`.
- [#668](#)³⁶⁴³: Allow line numbering of `code-block` and `literalinclude` directives to start at an arbitrary line number, with a new `lineno-start` option.
- [PR#172](#)³⁶⁴⁴, [PR#266](#)³⁶⁴⁵: The `code-block` and `literalinclude` directives now can have a `caption` option that shows a filename before the code in the output. Thanks to Nasimul Haque, Takeshi Komiya.
- Prompt for the document language in `sphinx-quickstart`.
- [PR#217](#)³⁶⁴⁶: Added config values to suppress UUID and location information in generated gettext catalogs.
- [PR#236](#)³⁶⁴⁷, [#1456](#)³⁶⁴⁸: `apidoc`: Add a `-M` option to put module documentation before submodule documentation. Thanks to Wes Turner and Luc Saffre.

³⁶³⁵ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#default-role>

³⁶³⁶ <https://github.com/sphinx-doc/sphinx/issues/214>

³⁶³⁷ <https://github.com/sphinx-doc/sphinx/issues/296>

³⁶³⁸ <https://github.com/sphinx-doc/sphinx/issues/303>

³⁶³⁹ <https://github.com/sphinx-doc/sphinx/issues/76>

³⁶⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/202>

³⁶⁴¹ <https://github.com/sphinx-doc/sphinx/issues/184>

³⁶⁴² <https://github.com/sphinx-doc/sphinx/issues/925>

³⁶⁴³ <https://github.com/sphinx-doc/sphinx/issues/668>

³⁶⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/172>

³⁶⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/266>

³⁶⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/217>

³⁶⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/236>

³⁶⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/1456>

- [#1434](#)³⁶⁴⁹: Provide non-minified JS files for jquery.js and underscore.js to clarify the source of the minified files.
- [PR#252](#)³⁶⁵⁰, [#1291](#)³⁶⁵¹: Windows color console support. Thanks to meu31.
- [PR#255](#)³⁶⁵²: When generating latex references, also insert latex target/anchor for the ids defined on the node. Thanks to Olivier Heurtier.
- [PR#229](#)³⁶⁵³: Allow registration of other translators. Thanks to Russell Sim.
- Add `app.set_translator()` API to register or override a Docutils translator class like `html_translator_class`.
- [PR#267](#)³⁶⁵⁴, [#1134](#)³⁶⁵⁵: add 'diff' parameter to `literalinclude`. Thanks to Richard Wall and WAKAYAMA shirou.
- [PR#272](#)³⁶⁵⁶: Added 'bizstyle' theme. Thanks to Shoji KUMAGAI.
- Automatically compile *.mo files from *.po files when `gettext_auto_build` is True (default) and *.po is newer than *.mo file.
- [#623](#)³⁶⁵⁷: `sphinx.ext.viewcode` supports imported function/class aliases.
- [PR#275](#)³⁶⁵⁸: `sphinx.ext.intersphinx` supports multiple target for the inventory. Thanks to Brigitta Sipocz.
- [PR#261](#)³⁶⁵⁹: Added the `env-before-read-docs` event that can be connected to modify the order of documents before they are read by the environment.
- [#1284](#)³⁶⁶⁰: Program options documented with *option* can now start with +.
- [PR#291](#)³⁶⁶¹: The caption of *code-block* is recognized as a title of ref target. Thanks to Takeshi Komiya.
- [PR#298](#)³⁶⁶²: Add new API: `add_latex_package()`. Thanks to Takeshi Komiya.
- [#1344](#)³⁶⁶³: add `gettext_enables` to enable extracting 'index' to gettext catalog output / applying translation catalog to generated documentation.
- [PR#301](#)³⁶⁶⁴, [#1583](#)³⁶⁶⁵: Allow the line numbering of the directive `literalinclude` to match that of the included file, using a new `lineno-match` option. Thanks to Jeppe Pihl.

³⁶⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/1434>

³⁶⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/252>

³⁶⁵¹ <https://github.com/sphinx-doc/sphinx/issues/1291>

³⁶⁵² <https://github.com/sphinx-doc/sphinx/issues/255>

³⁶⁵³ <https://github.com/sphinx-doc/sphinx/issues/229>

³⁶⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/267>

³⁶⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/1134>

³⁶⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/272>

³⁶⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/623>

³⁶⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/275>

³⁶⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/261>

³⁶⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/1284>

³⁶⁶¹ <https://github.com/sphinx-doc/sphinx/issues/291>

³⁶⁶² <https://github.com/sphinx-doc/sphinx/issues/298>

³⁶⁶³ <https://github.com/sphinx-doc/sphinx/issues/1344>

³⁶⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/301>

³⁶⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/1583>

- [PR#299](#)³⁶⁶⁶: add various options to sphinx-quickstart. Quiet mode option `--quiet` will skips wizard mode. Thanks to WAKAYAMA shirou.
- [#1623](#)³⁶⁶⁷: Return types specified with `:rtype:` are now turned into links if possible.

Bugs fixed

- [#1438](#)³⁶⁶⁸: Updated jQuery version from 1.8.3 to 1.11.1.
- [#1568](#)³⁶⁶⁹: Fix a crash when a "centered" directive contains a reference.
- Now sphinx.ext.autodoc works with python-2.5 again.
- [#1563](#)³⁶⁷⁰: `add_search_language()` raises `AssertionError` for correct type of argument. Thanks to rikoman.
- [#1174](#)³⁶⁷¹: Fix smart quotes being applied inside roles like `program` or `makevar`.
- [PR#235](#)³⁶⁷²: comment db schema of websupport lacked a length of the `node_id` field. Thanks to solos.
- [#1466](#)³⁶⁷³, [PR#241](#) <<https://github.com/sphinx-doc/sphinx/issues/241>>`: Fix failure of the cpp domain parser to parse C++ "variadic templates" declarations. Thanks to Victor Zverovich.
- [#1459](#)³⁶⁷⁴, [PR#244](#) <<https://github.com/sphinx-doc/sphinx/issues/244>>`: Fix default mathjax js path point to `http://` that cause mixed-content error on HTTPS server. Thanks to sbrandtb and robo9k.
- [PR#157](#)³⁶⁷⁵: autodoc remove spurious signatures from `@property` decorated attributes. Thanks to David Ham.
- [PR#159](#)³⁶⁷⁶: Add coverage targets to quickstart generated Makefile and make.bat. Thanks to Matthias Troffaes.
- [#1251](#)³⁶⁷⁷: When specifying `toctree :numbered:` option and `:tocdepth:` metadata, sub section number that is larger depth than `:tocdepth:` is shrunk.
- [PR#260](#)³⁶⁷⁸: Encode underscore in citation labels for latex export. Thanks to Lennart Fricke.

³⁶⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/299>

³⁶⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/1623>

³⁶⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/1438>

³⁶⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/1568>

³⁶⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/1563>

³⁶⁷¹ <https://github.com/sphinx-doc/sphinx/issues/1174>

³⁶⁷² <https://github.com/sphinx-doc/sphinx/issues/235>

³⁶⁷³ <https://github.com/sphinx-doc/sphinx/issues/1466>

³⁶⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/1459>

³⁶⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/157>

³⁶⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/159>

³⁶⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/1251>

³⁶⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/260>

- [PR#264³⁶⁷⁹](#): Fix could not resolve xref for figure node with :name: option. Thanks to Takeshi Komiya.
- [PR#265³⁶⁸⁰](#): Fix could not capture caption of graphviz node by xref. Thanks to Takeshi Komiya.
- [PR#263³⁶⁸¹](#), [#1013³⁶⁸²](#), [#1103³⁶⁸³](#): Rewrite of C++ domain. Thanks to Jakob Lykke Andersen.
 - Hyperlinks to all found nested names and template arguments ([#1103³⁶⁸⁴](#)).
 - Support for function types everywhere, e.g., in `std::function<bool(int, int)>` ([#1013³⁶⁸⁵](#)).
 - Support for virtual functions.
 - Changed interpretation of function arguments to following standard prototype declarations, i.e., `void f(arg)` means that `arg` is the type of the argument, instead of it being the name.
 - Updated tests.
 - Updated documentation with elaborate description of what declarations are supported and how the namespace declarations influence declaration and cross-reference lookup.
 - Index names may be different now. Elements are indexed by their fully qualified name. It should be rather easy to change this behaviour and potentially index by namespaces/classes as well.
- [PR#258³⁶⁸⁶](#), [#939³⁶⁸⁷](#): Add `dedent` option for `code-block` and `literalinclude`. Thanks to Zafar Siddiqui.
- [PR#268³⁶⁸⁸](#): Fix numbering section does not work at singlehtml mode. It still ad-hoc fix because there is a issue that section IDs are conflicted. Thanks to Takeshi Komiya.
- [PR#273³⁶⁸⁹](#), [#1536³⁶⁹⁰](#): Fix `RuntimeError` with numbered circular toctree. Thanks to Takeshi Komiya.
- [PR#274³⁶⁹¹](#): Set its URL as a default title value if URL appears in toctree. Thanks to Takeshi Komiya.

³⁶⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/264>

³⁶⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/265>

³⁶⁸¹ <https://github.com/sphinx-doc/sphinx/issues/263>

³⁶⁸² <https://github.com/sphinx-doc/sphinx/issues/1013>

³⁶⁸³ <https://github.com/sphinx-doc/sphinx/issues/1103>

³⁶⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/1103>

³⁶⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/1013>

³⁶⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/258>

³⁶⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/939>

³⁶⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/268>

³⁶⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/273>

³⁶⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/1536>

³⁶⁹¹ <https://github.com/sphinx-doc/sphinx/issues/274>

- [PR#276³⁶⁹²](#), [#1381³⁶⁹³](#): `rfc` and `pep` roles support custom link text. Thanks to Takeshi Komiya.
- [PR#277³⁶⁹⁴](#), [#1513³⁶⁹⁵](#): highlights for function pointers in argument list of `c:function`. Thanks to Takeshi Komiya.
- [PR#278³⁶⁹⁶](#): Fix section entries were shown twice if `toctree` has been put under only directive. Thanks to Takeshi Komiya.
- [#1547³⁶⁹⁷](#): `pgen2` tokenizer doesn't recognize `...` literal (Ellipsis for `py3`).
- [PR#294³⁶⁹⁸](#): On LaTeX builder, wrap float environment on writing `literal_block` to avoid separation of caption and body. Thanks to Takeshi Komiya.
- [PR#295³⁶⁹⁹](#), [#1520³⁷⁰⁰](#): `make.bat` `latexpdf` mechanism to `cd` back to the current directory. Thanks to Peter Suter.
- [PR#297³⁷⁰¹](#), [#1571³⁷⁰²](#): Add `imgpath` property to all builders. It make easier to develop builder extensions. Thanks to Takeshi Komiya.
- [#1584³⁷⁰³](#): Point to master doc in HTML "top" link.
- [#1585³⁷⁰⁴](#): Autosummary of modules broken in Sphinx 1.2.3.
- [#1610³⁷⁰⁵](#): Sphinx cause `AttributeError` when `MeCab` search option is enabled and `python-mecab` is not installed.
- [#1674³⁷⁰⁶](#): Do not crash if a module's `__all__` is not a list of strings.
- [#1673³⁷⁰⁷](#): Fix crashes with `nitpick_ignore` and `:doc:` references.
- [#1686³⁷⁰⁸](#): `ifconfig` directive doesn't care about default config values.
- [#1642³⁷⁰⁹](#): Fix only one search result appearing in Chrome.

³⁶⁹² <https://github.com/sphinx-doc/sphinx/issues/276>

³⁶⁹³ <https://github.com/sphinx-doc/sphinx/issues/1381>

³⁶⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/277>

³⁶⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/1513>

³⁶⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/278>

³⁶⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/1547>

³⁶⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/294>

³⁶⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/295>

³⁷⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/1520>

³⁷⁰¹ <https://github.com/sphinx-doc/sphinx/issues/297>

³⁷⁰² <https://github.com/sphinx-doc/sphinx/issues/1571>

³⁷⁰³ <https://github.com/sphinx-doc/sphinx/issues/1584>

³⁷⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/1585>

³⁷⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/1610>

³⁷⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/1674>

³⁷⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/1673>

³⁷⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/1686>

³⁷⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/1642>

Documentation

- Add clarification about the syntax of tags. (`doc/markup/misc.rst`)

14.144 Release 1.2.3 (released Sep 1, 2014)

Funzionalità aggiunte

- [#1518](#)³⁷¹⁰: `sphinx-apidoc` command now has a `--version` option to show version information and exit
- New locales: Hebrew, European Portuguese, Vietnamese.

Bugs fixed

- [#636](#)³⁷¹¹: Keep straight single quotes in literal blocks in the LaTeX build.
- [#1419](#)³⁷¹²: Generated `i18n sphinx.js` files are missing message catalog entries from `'js_'` and `'html'`. The issue was introduced from Sphinx 1.1
- [#1363](#)³⁷¹³: Fix `i18n`: missing python domain's cross-references with `currentmodule` directive or `currentclass` directive.
- [#1444](#)³⁷¹⁴: `autosummary` does not create the description from attributes docstring.
- [#1457](#)³⁷¹⁵: In python3 environment, make `linkcheck` cause "Can't convert 'bytes' object to str implicitly" error when link target url has a hash part. Thanks to Jorge_C.
- [#1467](#)³⁷¹⁶: Exception on Python3 if nonexistent method is specified by `automethod`
- [#1441](#)³⁷¹⁷: `autosummary` can't handle nested classes correctly.
- [#1499](#)³⁷¹⁸: With non-callable `setup` in a `conf.py`, now `sphinx-build` emits a user-friendly error message.
- [#1502](#)³⁷¹⁹: In `autodoc`, fix display of parameter defaults containing backslashes.
- [#1226](#)³⁷²⁰: `autodoc`, `autosummary`: importing `setup.py` by `automodule` will invoke `setup` process and execute `sys.exit()`. Now Sphinx avoids `SystemExit` exception and emits warnings without unexpected termination.

³⁷¹⁰ <https://github.com/sphinx-doc/sphinx/issues/1518>

³⁷¹¹ <https://github.com/sphinx-doc/sphinx/issues/636>

³⁷¹² <https://github.com/sphinx-doc/sphinx/issues/1419>

³⁷¹³ <https://github.com/sphinx-doc/sphinx/issues/1363>

³⁷¹⁴ <https://github.com/sphinx-doc/sphinx/issues/1444>

³⁷¹⁵ <https://github.com/sphinx-doc/sphinx/issues/1457>

³⁷¹⁶ <https://github.com/sphinx-doc/sphinx/issues/1467>

³⁷¹⁷ <https://github.com/sphinx-doc/sphinx/issues/1441>

³⁷¹⁸ <https://github.com/sphinx-doc/sphinx/issues/1499>

³⁷¹⁹ <https://github.com/sphinx-doc/sphinx/issues/1502>

³⁷²⁰ <https://github.com/sphinx-doc/sphinx/issues/1226>

- [#1503](#)³⁷²¹: `py:function` directive generate incorrectly signature when specifying a default parameter with an empty list `[]`. Thanks to Geert Jansen.
- [#1508](#)³⁷²²: Non-ASCII filename raise exception on `make singlehtml`, `latex`, `man`, `texinfo` and changes.
- [#1531](#)³⁷²³: On Python3 environment, `docutils.conf` with `'source_link=true'` in the general section cause type error.
- [PR#270](#)³⁷²⁴, [#1533](#)³⁷²⁵: Non-ASCII docstring cause `UnicodeDecodeError` when uses with `inheritance-diagram` directive. Thanks to WAKAYAMA shirou.
- [PR#281](#)³⁷²⁶, [PR#282](#)³⁷²⁷, [#1509](#)³⁷²⁸: `TODO` extension not compatible with `websupport`. Thanks to Takeshi Komiya.
- [#1477](#)³⁷²⁹: `gettext` does not extract `nodes.line` in a table or list.
- [#1544](#)³⁷³⁰: `make text` generates wrong table when it has empty table cells.
- [#1522](#)³⁷³¹: Footnotes from table get displayed twice in LaTeX. This problem has been appeared from Sphinx 1.2.1 by [#949](#)³⁷³².
- [#508](#)³⁷³³: Sphinx every time exit with zero when is invoked from `setup.py` command. ex. `python setup.py build_sphinx -b doctest` return zero even if doctest failed.

14.145 Release 1.2.2 (released Mar 2, 2014)

Bugs fixed

- [PR#211](#)³⁷³⁴: When checking for existence of the `html_logo` file, check the full relative path and not the basename.
- [PR#212](#)³⁷³⁵: Fix traceback with `autodoc` and `__init__` methods without docstring.
- [PR#213](#)³⁷³⁶: Fix a missing import in the `setup` command.

³⁷²¹ <https://github.com/sphinx-doc/sphinx/issues/1503>

³⁷²² <https://github.com/sphinx-doc/sphinx/issues/1508>

³⁷²³ <https://github.com/sphinx-doc/sphinx/issues/1531>

³⁷²⁴ <https://github.com/sphinx-doc/sphinx/issues/270>

³⁷²⁵ <https://github.com/sphinx-doc/sphinx/issues/1533>

³⁷²⁶ <https://github.com/sphinx-doc/sphinx/issues/281>

³⁷²⁷ <https://github.com/sphinx-doc/sphinx/issues/282>

³⁷²⁸ <https://github.com/sphinx-doc/sphinx/issues/1509>

³⁷²⁹ <https://github.com/sphinx-doc/sphinx/issues/1477>

³⁷³⁰ <https://github.com/sphinx-doc/sphinx/issues/1544>

³⁷³¹ <https://github.com/sphinx-doc/sphinx/issues/1522>

³⁷³² <https://github.com/sphinx-doc/sphinx/issues/949>

³⁷³³ <https://github.com/sphinx-doc/sphinx/issues/508>

³⁷³⁴ <https://github.com/sphinx-doc/sphinx/issues/211>

³⁷³⁵ <https://github.com/sphinx-doc/sphinx/issues/212>

³⁷³⁶ <https://github.com/sphinx-doc/sphinx/issues/213>

- [#1357](#)³⁷³⁷: Option names documented by *option* are now again allowed to not start with a dash or slash, and referencing them will work correctly.
- [#1358](#)³⁷³⁸: Fix handling of image paths outside of the source directory when using the "wildcard" style reference.
- [#1374](#)³⁷³⁹: Fix for autosummary generating overly-long summaries if first line doesn't end with a period.
- [#1383](#)³⁷⁴⁰: Fix Python 2.5 compatibility of sphinx-apidoc.
- [#1391](#)³⁷⁴¹: Actually prevent using "pngmath" and "mathjax" extensions at the same time in sphinx-quickstart.
- [#1386](#)³⁷⁴²: Fix bug preventing more than one theme being added by the entry point mechanism.
- [#1370](#)³⁷⁴³: Ignore "toctree" nodes in text writer, instead of raising.
- [#1364](#)³⁷⁴⁴: Fix 'make gettext' fails when the '.. todolist:.' directive is present.
- [#1367](#)³⁷⁴⁵: Fix a change of [PR#96](#)³⁷⁴⁶ that break sphinx.util.docfields.Field.make_field interface/behavior for item argument usage.

Documentation

- Extended the *documentation about building extensions*.

14.146 Release 1.2.1 (released Jan 19, 2014)

Bugs fixed

- [#1335](#)³⁷⁴⁷: Fix autosummary template overloading with exclamation prefix like `{% extends "!autosummary/class.rst" %}` cause infinite recursive function call. This was caused by [PR#181](#)³⁷⁴⁸.

³⁷³⁷ <https://github.com/sphinx-doc/sphinx/issues/1357>

³⁷³⁸ <https://github.com/sphinx-doc/sphinx/issues/1358>

³⁷³⁹ <https://github.com/sphinx-doc/sphinx/issues/1374>

³⁷⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/1383>

³⁷⁴¹ <https://github.com/sphinx-doc/sphinx/issues/1391>

³⁷⁴² <https://github.com/sphinx-doc/sphinx/issues/1386>

³⁷⁴³ <https://github.com/sphinx-doc/sphinx/issues/1370>

³⁷⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/1364>

³⁷⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/1367>

³⁷⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/96>

³⁷⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/1335>

³⁷⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/181>

- #1337³⁷⁴⁹: Fix autodoc with `autoclass_content="both"` uses useless object. `__init__` docstring when class does not have `__init__`. This was caused by a change for #1138³⁷⁵⁰.
- #1340³⁷⁵¹: Can't search alphabetical words on the HTML quick search generated with `language='ja'`.
- #1319³⁷⁵²: Do not crash if the `html_logo` file does not exist.
- #603³⁷⁵³: Do not use the HTML-ized title for building the search index (that resulted in "literal" being found on every page with a literal in the title).
- #751³⁷⁵⁴: Allow production lists longer than a page in LaTeX by using `longtable`.
- #764³⁷⁵⁵: Always look for stopwords lowercased in JS search.
- #814³⁷⁵⁶: autodoc: Guard against strange type objects that don't have `__bases__`.
- #932³⁷⁵⁷: autodoc: Do not crash if `__doc__` is not a string.
- #933³⁷⁵⁸: Do not crash if an `option` value is malformed (contains spaces but no option name).
- #908³⁷⁵⁹: On Python 3, handle error messages from LaTeX correctly in the `pngmath` extension.
- #943³⁷⁶⁰: In `autosummary`, recognize "first sentences" to pull from the docstring if they contain uppercase letters.
- #923³⁷⁶¹: Take the entire LaTeX document into account when caching `pngmath`-generated images. This rebuilds them correctly when `pngmath_latex_preamble` changes.
- #901³⁷⁶²: Emit a warning when using Docutils' new "math" markup without a Sphinx math extension active.
- #845³⁷⁶³: In code blocks, when the selected lexer fails, display line numbers nevertheless if configured.
- #929³⁷⁶⁴: Support parsed-literal blocks in LaTeX output correctly.
- #949³⁷⁶⁵: Update the `tabulary.sty` packed with Sphinx.

³⁷⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/1337>

³⁷⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/1138>

³⁷⁵¹ <https://github.com/sphinx-doc/sphinx/issues/1340>

³⁷⁵² <https://github.com/sphinx-doc/sphinx/issues/1319>

³⁷⁵³ <https://github.com/sphinx-doc/sphinx/issues/603>

³⁷⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/751>

³⁷⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/764>

³⁷⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/814>

³⁷⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/932>

³⁷⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/933>

³⁷⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/908>

³⁷⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/943>

³⁷⁶¹ <https://github.com/sphinx-doc/sphinx/issues/923>

³⁷⁶² <https://github.com/sphinx-doc/sphinx/issues/901>

³⁷⁶³ <https://github.com/sphinx-doc/sphinx/issues/845>

³⁷⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/929>

³⁷⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/949>

- [#1050³⁷⁶⁶](#): Add anonymous labels into `objects.inv` to be referenced via *intersphinx*.
- [#1095³⁷⁶⁷](#): Fix print-media stylesheet being included always in the "scrolls" theme.
- [#1085³⁷⁶⁸](#): Fix current classname not getting set if class description has `:noindex:` set.
- [#1181³⁷⁶⁹](#): Report option errors in autodoc directives more gracefully.
- [#1155³⁷⁷⁰](#): Fix aut documenting C-defined methods as attributes in Python 3.
- [#1233³⁷⁷¹](#): Allow finding both Python classes and exceptions with the "class" and "exc" roles in intersphinx.
- [#1198³⁷⁷²](#): Allow "image" for the "figwidth" option of the *figure*³⁷⁷³ directive as documented by docutils.
- [#1152³⁷⁷⁴](#): Fix pycode parsing errors of Python 3 code by including two grammar versions for Python 2 and 3, and loading the appropriate version for the running Python version.
- [#1017³⁷⁷⁵](#): Be helpful and tell the user when the argument to *option* does not match the required format.
- [#1345³⁷⁷⁶](#): Fix two bugs with `nitpick_ignore`; now you don't have to remove the store environment for changes to have effect.
- [#1072³⁷⁷⁷](#): In the JS search, fix issues searching for upper-cased words by lowercasing words before stemming.
- [#1299³⁷⁷⁸](#): Make behavior of the *math* directive more consistent and avoid producing empty environments in LaTeX output.
- [#1308³⁷⁷⁹](#): Strip HTML tags from the content of "raw" nodes before feeding it to the search indexer.
- [#1249³⁷⁸⁰](#): Fix duplicate LaTeX page numbering for manual documents.
- [#1292³⁷⁸¹](#): In the linkchecker, retry HEAD requests when denied by HTTP 405. Also make the redirect code apparent and tweak the output a bit to be more obvious.
- [#1285³⁷⁸²](#): Avoid name clashes between C domain objects and section titles.

³⁷⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/1050>

³⁷⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/1095>

³⁷⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/1085>

³⁷⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/1181>

³⁷⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/1155>

³⁷⁷¹ <https://github.com/sphinx-doc/sphinx/issues/1233>

³⁷⁷² <https://github.com/sphinx-doc/sphinx/issues/1198>

³⁷⁷³ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#figure>

³⁷⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/1152>

³⁷⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/1017>

³⁷⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/1345>

³⁷⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/1072>

³⁷⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/1299>

³⁷⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/1308>

³⁷⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/1249>

³⁷⁸¹ <https://github.com/sphinx-doc/sphinx/issues/1292>

³⁷⁸² <https://github.com/sphinx-doc/sphinx/issues/1285>

- #848³⁷⁸³: Always take the newest code in incremental rebuilds with the `sphinx.ext.viewcode` extension.
- #979³⁷⁸⁴, #1266³⁷⁸⁵: Fix exclude handling in sphinx-apidoc.
- #1302³⁷⁸⁶: Fix regression in `sphinx.ext.inheritance_diagram` when documenting classes that can't be pickled.
- #1316³⁷⁸⁷: Remove hard-coded font-face resources from epub theme.
- #1329³⁷⁸⁸: Fix traceback with empty translation msgstr in .po files.
- #1300³⁷⁸⁹: Fix references not working in translated documents in some instances.
- #1283³⁷⁹⁰: Fix a bug in the detection of changed files that would try to access doctrees of deleted documents.
- #1330³⁷⁹¹: Fix `exclude_patterns` behavior with subdirectories in the `html_static_path`.
- #1323³⁷⁹²: Fix emitting empty `` tags in the HTML writer, which is not valid HTML.
- #1147³⁷⁹³: Don't emit a sidebar search box in the "singlehtml" builder.

Documentation

- #1325³⁷⁹⁴: Added a "Intersphinx" tutorial section. (`doc/tutorial.rst`)

14.147 Release 1.2 (released Dec 10, 2013)

Funzionalità aggiunte

- Added `sphinx.version_info` tuple for programmatic checking of the Sphinx version.

³⁷⁸³ <https://github.com/sphinx-doc/sphinx/issues/848>

³⁷⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/979>

³⁷⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/1266>

³⁷⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/1302>

³⁷⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/1316>

³⁷⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/1329>

³⁷⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/1300>

³⁷⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/1283>

³⁷⁹¹ <https://github.com/sphinx-doc/sphinx/issues/1330>

³⁷⁹² <https://github.com/sphinx-doc/sphinx/issues/1323>

³⁷⁹³ <https://github.com/sphinx-doc/sphinx/issues/1147>

³⁷⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/1325>

Modifiche incompatibili

- Removed the `sphinx.ext.refcounting` extension -- it is very specific to CPython and has no place in the main distribution.

Bugs fixed

- Restore `versionmodified` CSS class for `versionadded/changed` and `deprecated` directives.
- [PR#181](#)³⁷⁹⁵: Fix `html_theme_path = ['.']` is a trigger of rebuild all documents always (This change keeps the current "theme changes cause a rebuild" feature).
- [#1296](#)³⁷⁹⁶: Fix invalid charset in HTML help generated HTML files for default locale.
- [PR#190](#)³⁷⁹⁷: Fix `gettext` does not extract figure caption and rubric title inside other blocks. Thanks to Michael Schlenker.
- [PR#176](#)³⁷⁹⁸: Make sure `setup_command` test can always import Sphinx. Thanks to Dmitry Shachnev.
- [#1311](#)³⁷⁹⁹: Fix `test_linkcode.test_html` fails with C locale and Python 3.
- [#1269](#)³⁸⁰⁰: Fix `ResourceWarnings` with Python 3.2 or later.
- [#1138](#)³⁸⁰¹: Fix: When `autodoc_docstring_signature = True` and `autoclass_content = 'init'` or `'both'`, `__init__` line should be removed from class documentation.

14.148 Release 1.2 beta3 (released Oct 3, 2013)

Funzionalità aggiunte

- The Sphinx error log files will now include a list of the loaded extensions for help in debugging.

³⁷⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/181>

³⁷⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/1296>

³⁷⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/190>

³⁷⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/176>

³⁷⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/1311>

³⁸⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/1269>

³⁸⁰¹ <https://github.com/sphinx-doc/sphinx/issues/1138>

Modifiche incompatibili

- [PR#154](#)³⁸⁰²: Remove "sphinx" prefix from LaTeX class name except 'sphinxmanual' and 'sphinxhowto'. Now you can use your custom document class without 'sphinx' prefix. Thanks to Erik B.

Bugs fixed

- [#1265](#)³⁸⁰³: Fix i18n: crash when translating a section name that is pointed to from a named target.
- A wrong condition broke the search feature on first page that is usually index.rst. This issue was introduced in 1.2b1.
- [#703](#)³⁸⁰⁴: When Sphinx can't decode filenames with non-ASCII characters, Sphinx now catches UnicodeError and will continue if possible instead of raising the exception.

14.149 Release 1.2 beta2 (released Sep 17, 2013)

Funzionalità aggiunte

- apidoc now ignores "_private" modules by default, and has an option -P to include them.
- apidoc now has an option to not generate headings for packages and modules, for the case that the module docstring already includes a reST heading.
- [PR#161](#)³⁸⁰⁵: apidoc can now write each module to a standalone page instead of combining all modules in a package on one page.
- Builders: rebuild i18n target document when catalog updated.
- Support docutils.conf 'writers' and 'html4css1 writer' section in the HTML writer. The latex, manpage and texinfo writers also support their respective 'writers' sections.
- The new `html_extra_path` config value allows to specify directories with files that should be copied directly to the HTML output directory.
- Autodoc directives for module data and attributes now support an `annotation` option, so that the default display of the data/attribute value can be overridden.
- [PR#136](#)³⁸⁰⁶: Autodoc directives now support an `imported-members` option to include members imported from different modules.
- New locales: Macedonian, Sinhala, Indonesian.

³⁸⁰² <https://github.com/sphinx-doc/sphinx/issues/154>

³⁸⁰³ <https://github.com/sphinx-doc/sphinx/issues/1265>

³⁸⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/703>

³⁸⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/161>

³⁸⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/136>

- Theme package collection by using setuptools plugin mechanism.

Modifiche incompatibili

- [PR#144](#)³⁸⁰⁷, [#1182](#)³⁸⁰⁸: Force timezone offset to LocalTimeZone on POT-Creation-Date that was generated by gettext builder. Thanks to masklinn and Jakub Wilk.

Bugs fixed

- [PR#132](#)³⁸⁰⁹: Updated jQuery version to 1.8.3.
- [PR#141](#)³⁸¹⁰, [#982](#)³⁸¹¹: Avoid crash when writing PNG file using Python 3. Thanks to Marcin Wojdyr.
- [PR#145](#)³⁸¹²: In parallel builds, Sphinx drops second document file to write. Thanks to tychoish.
- [PR#151](#)³⁸¹³: Some styling updates to tables in LaTeX.
- [PR#153](#)³⁸¹⁴: The "extensions" config value can now be overridden.
- [PR#155](#)³⁸¹⁵: Added support for some C++11 function qualifiers.
- Fix: 'make gettext' caused UnicodeDecodeError when templates contain utf-8 encoded strings.
- [#828](#)³⁸¹⁶: use inspect.getfullargspec() to be able to document functions with keyword-only arguments on Python 3.
- [#1090](#)³⁸¹⁷: Fix i18n: multiple cross references (term, ref, doc) in the same line return the same link.
- [#1157](#)³⁸¹⁸: Combination of 'globaltoc.html' and hidden toctree caused exception.
- [#1159](#)³⁸¹⁹: fix wrong generation of objects inventory for Python modules, and add a workaround in intersphinx to fix handling of affected inventories.
- [#1160](#)³⁸²⁰: Citation target missing caused an AssertionError.

³⁸⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/144>

³⁸⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/1182>

³⁸⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/132>

³⁸¹⁰ <https://github.com/sphinx-doc/sphinx/issues/141>

³⁸¹¹ <https://github.com/sphinx-doc/sphinx/issues/982>

³⁸¹² <https://github.com/sphinx-doc/sphinx/issues/145>

³⁸¹³ <https://github.com/sphinx-doc/sphinx/issues/151>

³⁸¹⁴ <https://github.com/sphinx-doc/sphinx/issues/153>

³⁸¹⁵ <https://github.com/sphinx-doc/sphinx/issues/155>

³⁸¹⁶ <https://github.com/sphinx-doc/sphinx/issues/828>

³⁸¹⁷ <https://github.com/sphinx-doc/sphinx/issues/1090>

³⁸¹⁸ <https://github.com/sphinx-doc/sphinx/issues/1157>

³⁸¹⁹ <https://github.com/sphinx-doc/sphinx/issues/1159>

³⁸²⁰ <https://github.com/sphinx-doc/sphinx/issues/1160>

- [#1162](#)³⁸²¹, [PR#139](#)³⁸²²: singlehtml builder didn't copy images to `_images/`.
- [#1173](#)³⁸²³: Adjust `setup.py` dependencies because Jinja2 2.7 discontinued compatibility with Python < 3.3 and Python < 2.6. Thanks to Alexander Dupuy.
- [#1185](#)³⁸²⁴: Don't crash when a Python module has a wrong or no encoding declared, and non-ASCII characters are included.
- [#1188](#)³⁸²⁵: sphinx-quickstart raises `UnicodeEncodeError` if "Project version" includes non-ASCII characters.
- [#1189](#)³⁸²⁶: "Title underline is too short" WARNING is given when using fullwidth characters to "Project name" on quickstart.
- [#1190](#)³⁸²⁷: Output TeX/texinfo/man filename has no basename (only extension) when using non-ASCII characters in the "Project name" on quickstart.
- [#1192](#)³⁸²⁸: Fix escaping problem for hyperlinks in the manpage writer.
- [#1193](#)³⁸²⁹: Fix i18n: multiple link references in the same line return the same link.
- [#1176](#)³⁸³⁰: Fix i18n: footnote reference number missing for auto numbered named footnote and auto symbol footnote.
- [PR#146](#)³⁸³¹, [#1172](#) <<https://github.com/sphinx-doc/sphinx/issues/1172>>`: Fix `ZeroDivisionError` in parallel builds. Thanks to tychoish.
- [#1204](#)³⁸³²: Fix wrong generation of links to local intersphinx targets.
- [#1206](#)³⁸³³: Fix i18n: gettext did not translate admonition directive's title.
- [#1232](#)³⁸³⁴: Sphinx generated broken ePub files on Windows.
- [#1259](#)³⁸³⁵: Guard the debug output call when emitting events; to prevent the `repr()` implementation of arbitrary objects causing build failures.
- [#1142](#)³⁸³⁶: Fix NFC/NFD normalizing problem of rst filename on Mac OS X.
- [#1234](#)³⁸³⁷: Ignoring the string consists only of white-space characters.

³⁸²¹ <https://github.com/sphinx-doc/sphinx/issues/1162>

³⁸²² <https://github.com/sphinx-doc/sphinx/issues/139>

³⁸²³ <https://github.com/sphinx-doc/sphinx/issues/1173>

³⁸²⁴ <https://github.com/sphinx-doc/sphinx/issues/1185>

³⁸²⁵ <https://github.com/sphinx-doc/sphinx/issues/1188>

³⁸²⁶ <https://github.com/sphinx-doc/sphinx/issues/1189>

³⁸²⁷ <https://github.com/sphinx-doc/sphinx/issues/1190>

³⁸²⁸ <https://github.com/sphinx-doc/sphinx/issues/1192>

³⁸²⁹ <https://github.com/sphinx-doc/sphinx/issues/1193>

³⁸³⁰ <https://github.com/sphinx-doc/sphinx/issues/1176>

³⁸³¹ <https://github.com/sphinx-doc/sphinx/issues/146>

³⁸³² <https://github.com/sphinx-doc/sphinx/issues/1204>

³⁸³³ <https://github.com/sphinx-doc/sphinx/issues/1206>

³⁸³⁴ <https://github.com/sphinx-doc/sphinx/issues/1232>

³⁸³⁵ <https://github.com/sphinx-doc/sphinx/issues/1259>

³⁸³⁶ <https://github.com/sphinx-doc/sphinx/issues/1142>

³⁸³⁷ <https://github.com/sphinx-doc/sphinx/issues/1234>

14.150 Release 1.2 beta1 (released Mar 31, 2013)

Modifiche incompatibili

- Removed `sphinx.util.compat.directive_dwim()` and `sphinx.roles.xfileref_role()` which were deprecated since version 1.0.
- [PR#122³⁸³⁸](https://github.com/sphinx-doc/sphinx/issues/122): the files given in `latex_additional_files` now override TeX files included by Sphinx, such as `sphinx.sty`.
- [PR#124³⁸³⁹](https://github.com/sphinx-doc/sphinx/issues/124): the node generated by `versionadded`, `versionchanged` and deprecated directives now includes all added markup (such as "New in version X") as child nodes, and no additional text must be generated by writers.
- [PR#99³⁸⁴⁰](https://github.com/sphinx-doc/sphinx/issues/99): the `seealso` directive now generates admonition nodes instead of the custom `seealso` node.

Funzionalità aggiunte

- Markup
 - The `toctree` directive and the `toctree()` template function now have an `includehidden` option that includes hidden toctree entries (bugs [#790³⁸⁴¹](https://github.com/sphinx-doc/sphinx/issues/790) and [#1047³⁸⁴²](https://github.com/sphinx-doc/sphinx/issues/1047)). A bug in the `maxdepth` option for the `toctree()` template function has been fixed (bug [#1046³⁸⁴³](https://github.com/sphinx-doc/sphinx/issues/1046)).
 - [PR#99³⁸⁴⁴](https://github.com/sphinx-doc/sphinx/issues/99): Strip down `seealso` directives to normal admonitions. This removes their unusual CSS classes (`admonition-see-also`), inconsistent LaTeX admonition title ("See Also" instead of "See also"), and spurious indentation in the text builder.
- HTML builder
 - [#783³⁸⁴⁵](https://github.com/sphinx-doc/sphinx/issues/783): Create a link to full size image if it is scaled with width or height.
 - [#1067³⁸⁴⁶](https://github.com/sphinx-doc/sphinx/issues/1067): Improve the ordering of the JavaScript search results: matches in titles come before matches in full text, and object results are better categorized. Also implement a pluggable search scorer.
 - [#1053³⁸⁴⁷](https://github.com/sphinx-doc/sphinx/issues/1053): The "rightsidebar" and "collapsiblesidebar" HTML theme options now work together.

³⁸³⁸ <https://github.com/sphinx-doc/sphinx/issues/122>

³⁸³⁹ <https://github.com/sphinx-doc/sphinx/issues/124>

³⁸⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/99>

³⁸⁴¹ <https://github.com/sphinx-doc/sphinx/issues/790>

³⁸⁴² <https://github.com/sphinx-doc/sphinx/issues/1047>

³⁸⁴³ <https://github.com/sphinx-doc/sphinx/issues/1046>

³⁸⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/99>

³⁸⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/783>

³⁸⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/1067>

³⁸⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/1053>

- Update to jQuery 1.7.1 and Underscore.js 1.3.1.
- Texinfo builder
 - An "Index" node is no longer added when there are no entries.
 - "deffn" categories are no longer capitalized if they contain capital letters.
 - desc_annotation nodes are now rendered.
 - strong and emphasis nodes are now formatted like literals. The reason for this is because the standard Texinfo markup (`*strong*` and `_emphasis_`) resulted in confusing output due to the common usage of using these constructs for documenting parameter names.
 - Field lists formatting has been tweaked to better display "Info field lists".
 - system_message and problematic nodes are now formatted in a similar fashion as done by the text builder.
 - "en-dash" and "em-dash" conversion of hyphens is no longer performed in option directive signatures.
 - @ref is now used instead of @pxref for cross-references which prevents the word "see" from being added before the link (does not affect the Info output).
 - The @finalout command has been added for better TeX output.
 - transition nodes are now formatted using underscores ("_") instead of asterisks ("*").
 - The default value for the paragraphindent has been changed from 2 to 0 meaning that paragraphs are no longer indented by default.
 - #1110³⁸⁴⁸: A new configuration value texinfo_no_detailmenu has been added for controlling whether a @detailmenu is added in the "Top" node's menu.
 - Detailed menus are no longer created except for the "Top" node.
 - Fixed an issue where duplicate domain indices would result in invalid output.
- LaTeX builder:
 - PR#115³⁸⁴⁹: Add 'transition' item in latex_elements for customizing how transitions are displayed. Thanks to Jeff Klukas.
 - PR#114³⁸⁵⁰: The LaTeX writer now includes the "cmap" package by default. The 'cmappkg' item in latex_elements can be used to control this. Thanks to Dmitry Shachnev.
 - The 'fontpkg' item in latex_elements now defaults to '' when the language uses the Cyrillic script. Suggested by Dmitry Shachnev.
 - The latex_documents, texinfo_documents, and man_pages configuration values will be set to default values based on the master_doc if not explicitly set in

³⁸⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/1110>

³⁸⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/115>

³⁸⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/114>

`conf.py`. Previously, if these values were not set, no output would be generated by their respective builders.

- Internationalization:
 - Add i18n capabilities for custom templates. For example: The Sphinx reference documentation in `doc` directory provides a `sphinx.pot` file with message strings from `doc/_templates/*.html` when using `make gettext`.
 - [PR#61³⁸⁵¹, #703 <https://github.com/sphinx-doc/sphinx/issues/703>](https://github.com/sphinx-doc/sphinx/issues/703)³⁸⁵¹: Add support for non-ASCII filename handling.
- Other builders:
 - Added the Docutils-native XML and pseudo-XML builders. See *XMLBuilder* and *PseudoXMLBuilder*.
 - [PR#45³⁸⁵²](https://github.com/sphinx-doc/sphinx/issues/45): The linkcheck builder now checks `#anchors` for existence.
 - [PR#123³⁸⁵³](https://github.com/sphinx-doc/sphinx/issues/123), [#1106³⁸⁵⁴](https://github.com/sphinx-doc/sphinx/issues/1106): Add `epub_use_index` configuration value. If provided, it will be used instead of `html_use_index` for epub builder.
 - [PR#126³⁸⁵⁵](https://github.com/sphinx-doc/sphinx/issues/126): Add `epub_tocscope` configuration value. The setting controls the generation of the epub toc. The user can now also include hidden toc entries.
 - [PR#112³⁸⁵⁶](https://github.com/sphinx-doc/sphinx/issues/112): Add `epub_show_urls` configuration value.
- Extensions:
 - [PR#52³⁸⁵⁷](https://github.com/sphinx-doc/sphinx/issues/52): `special_members` flag to autodoc now behaves like `members`.
 - [PR#47³⁸⁵⁸](https://github.com/sphinx-doc/sphinx/issues/47): Added *sphinx.ext.linkcode* extension.
 - [PR#25³⁸⁵⁹](https://github.com/sphinx-doc/sphinx/issues/25): In inheritance diagrams, the first line of the class docstring is now the tooltip for the class.
- Command-line interfaces:
 - [PR#75³⁸⁶⁰](https://github.com/sphinx-doc/sphinx/issues/75): Added `--follow-links` option to `sphinx-apidoc`.
 - [#869³⁸⁶¹](https://github.com/sphinx-doc/sphinx/issues/869): `sphinx-build` now has the option `-T` for printing the full traceback after an unhandled exception.
 - `sphinx-build` now supports the standard `--help` and `--version` options.
 - `sphinx-build` now provides more specific error messages when called with invalid options or arguments.

³⁸⁵¹ <https://github.com/sphinx-doc/sphinx/issues/61>

³⁸⁵² <https://github.com/sphinx-doc/sphinx/issues/45>

³⁸⁵³ <https://github.com/sphinx-doc/sphinx/issues/123>

³⁸⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/1106>

³⁸⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/126>

³⁸⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/112>

³⁸⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/52>

³⁸⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/47>

³⁸⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/25>

³⁸⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/75>

³⁸⁶¹ <https://github.com/sphinx-doc/sphinx/issues/869>

- sphinx-build now has a verbose option `-v` which can be repeated for greater effect. A single occurrence provides a slightly more verbose output than normal. Two or more occurrences of this option provides more detailed output which may be useful for debugging.
- Locales:
 - [PR#74](#)³⁸⁶²: Fix some Russian translation.
 - [PR#54](#)³⁸⁶³: Added Norwegian bokmaal translation.
 - [PR#35](#)³⁸⁶⁴: Added Slovak translation.
 - [PR#28](#)³⁸⁶⁵: Added Hungarian translation.
 - [#1113](#)³⁸⁶⁶: Add Hebrew locale.
 - [#1097](#)³⁸⁶⁷: Add Basque locale.
 - [#1037](#)³⁸⁶⁸: Fix typos in Polish translation. Thanks to Jakub Wilk.
 - [#1012](#)³⁸⁶⁹: Update Estonian translation.
- Optimizations:
 - Speed up building the search index by caching the results of the word stemming routines. Saves about 20 seconds when building the Python documentation.
 - [PR#108](#)³⁸⁷⁰: Add experimental support for parallel building with a new *sphinx-build -j* option.

Documentation

- [PR#88](#)³⁸⁷¹: Added the "Sphinx Developer's Guide" (`doc/devguide.rst`) which outlines the basic development process of the Sphinx project.
- Added a detailed "Installing Sphinx" document (`doc/install.rst`).

³⁸⁶² <https://github.com/sphinx-doc/sphinx/issues/74>

³⁸⁶³ <https://github.com/sphinx-doc/sphinx/issues/54>

³⁸⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/35>

³⁸⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/28>

³⁸⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/1113>

³⁸⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/1097>

³⁸⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/1037>

³⁸⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/1012>

³⁸⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/108>

³⁸⁷¹ <https://github.com/sphinx-doc/sphinx/issues/88>

Bugs fixed

- [PR#124³⁸⁷²](#): Fix paragraphs in versionmodified are ignored when it has no dangling paragraphs. Fix wrong html output (nested <p> tag). Fix versionmodified is not translatable. Thanks to Nozomu Kaneko.
- [PR#111³⁸⁷³](#): Respect `add_autodoc_attrgetter()` even when `inherited-members` is set. Thanks to A. Jesse Jiryu Davis.
- [PR#97³⁸⁷⁴](#): Fix footnote handling in translated documents.
- Fix text writer not handling `visit_legend` for figure directive contents.
- Fix text builder not respecting wide/fullwidth characters: title underline width, table layout width and text wrap width.
- Fix leading space in LaTeX table header cells.
- [#1132³⁸⁷⁵](#): Fix LaTeX table output for multi-row cells in the first column.
- [#1128³⁸⁷⁶](#): Fix Unicode errors when trying to format time strings with a non-standard locale.
- [#1127³⁸⁷⁷](#): Fix traceback when autodoc tries to tokenize a non-Python file.
- [#1126³⁸⁷⁸](#): Fix double-hyphen to en-dash conversion in wrong places such as command-line option names in LaTeX.
- [#1123³⁸⁷⁹](#): Allow whitespaces in filenames given to `literalinclude`.
- [#1120³⁸⁸⁰](#): Added improvements about `i18n` for themes "basic", "haiku" and "scrolls" that Sphinx built-in. Thanks to Leonardo J. Caballero G.
- [#1118³⁸⁸¹](#): Updated Spanish translation. Thanks to Leonardo J. Caballero G.
- [#1117³⁸⁸²](#): Handle `.pyx` files in `sphinx-apidoc`.
- [#1112³⁸⁸³](#): Avoid duplicate download files when referenced from documents in different ways (absolute/relative).
- [#1111³⁸⁸⁴](#): Fix failure to find uppercase words in search when `html_search_language` is 'ja'. Thanks to Tomo Saito.
- [#1108³⁸⁸⁵](#): The text writer now correctly numbers enumerated lists with non-default start

³⁸⁷² <https://github.com/sphinx-doc/sphinx/issues/124>

³⁸⁷³ <https://github.com/sphinx-doc/sphinx/issues/111>

³⁸⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/97>

³⁸⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/1132>

³⁸⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/1128>

³⁸⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/1127>

³⁸⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/1126>

³⁸⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/1123>

³⁸⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/1120>

³⁸⁸¹ <https://github.com/sphinx-doc/sphinx/issues/1118>

³⁸⁸² <https://github.com/sphinx-doc/sphinx/issues/1117>

³⁸⁸³ <https://github.com/sphinx-doc/sphinx/issues/1112>

³⁸⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/1111>

³⁸⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/1108>

values (based on patch by Ewan Edwards).

- [#1102³⁸⁸⁶](#): Support multi-context "with" statements in autodoc.
- [#1090³⁸⁸⁷](#): Fix gettext not extracting glossary terms.
- [#1074³⁸⁸⁸](#): Add environment version info to the generated search index to avoid compatibility issues with old builds.
- [#1070³⁸⁸⁹](#): Avoid un-pickling issues when running Python 3 and the saved environment was created under Python 2.
- [#1069³⁸⁹⁰](#): Fixed error caused when autodoc would try to format signatures of "partial" functions without keyword arguments (patch by Artur Gaspar).
- [#1062³⁸⁹¹](#): sphinx.ext.autodoc use `__init__` method signature for class signature.
- [#1055³⁸⁹²](#): Fix web support with relative path to source directory.
- [#1043³⁸⁹³](#): Fix sphinx-quickstart asking again for yes/no questions because `input()` returns values with an extra 'r' on Python 3.2.0 + Windows. Thanks to Régis Décamps.
- [#1041³⁸⁹⁴](#): Fix failure of the cpp domain parser to parse a const type with a modifier.
- [#1038³⁸⁹⁵](#): Fix failure of the cpp domain parser to parse C++11 "static constexpr" declarations. Thanks to Jakub Wilk.
- [#1029³⁸⁹⁶](#): Fix intersphinx_mapping values not being stable if the mapping has plural key/value set with Python 3.3.
- [#1028³⁸⁹⁷](#): Fix line block output in the text builder.
- [#1024³⁸⁹⁸](#): Improve Makefile/make.bat error message if Sphinx is not found. Thanks to Anatoly Techtonik.
- [#1018³⁸⁹⁹](#): Fix "container" directive handling in the text builder.
- [#1015³⁹⁰⁰](#): Stop overriding jQuery contains() in the JavaScript.
- [#1010³⁹⁰¹](#): Make pngmath images transparent by default; IE7+ should handle it.
- [#1008³⁹⁰²](#): Fix test failures with Python 3.3.

³⁸⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/1102>

³⁸⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/1090>

³⁸⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/1074>

³⁸⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/1070>

³⁸⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/1069>

³⁸⁹¹ <https://github.com/sphinx-doc/sphinx/issues/1062>

³⁸⁹² <https://github.com/sphinx-doc/sphinx/issues/1055>

³⁸⁹³ <https://github.com/sphinx-doc/sphinx/issues/1043>

³⁸⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/1041>

³⁸⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/1038>

³⁸⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/1029>

³⁸⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/1028>

³⁸⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/1024>

³⁸⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/1018>

³⁹⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/1015>

³⁹⁰¹ <https://github.com/sphinx-doc/sphinx/issues/1010>

³⁹⁰² <https://github.com/sphinx-doc/sphinx/issues/1008>

- [#995³⁹⁰³](#): Fix table-of-contents and page numbering for the LaTeX "howto" class.
- [#976³⁹⁰⁴](#): Fix gettext does not extract index entries.
- [PR#72³⁹⁰⁵](#): [#975³⁹⁰⁶](#): Fix gettext not extracting definition terms before Docutils 0.10.
- [#961³⁹⁰⁷](#): Fix LaTeX output for triple quotes in code snippets.
- [#958³⁹⁰⁸](#): Do not preserve `environment.pickle` after a failed build.
- [#955³⁹⁰⁹](#): Fix `i18n` transformation.
- [#940³⁹¹⁰](#): Fix gettext does not extract figure caption.
- [#920³⁹¹¹](#): Fix PIL packaging issue that allowed to import Image without PIL namespace. Thanks to Marc Schlaich.
- [#723³⁹¹²](#): Fix the search function on local files in WebKit based browsers.
- [#440³⁹¹³](#): Fix coarse timestamp resolution in some filesystem generating a wrong list of outdated files.

14.151 Release 1.1.3 (Mar 10, 2012)

- [PR#40³⁹¹⁴](#): Fix `safe_repr` function to decode bytestrings with non-ASCII characters correctly.
- [PR#37³⁹¹⁵](#): Allow configuring sphinx-apidoc via `SPHINX_APIDOC_OPTIONS`.
- [PR#34³⁹¹⁶](#): Restore Python 2.4 compatibility.
- [PR#36³⁹¹⁷](#): Make the "bibliography to TOC" fix in LaTeX output specific to the document class.
- [#695³⁹¹⁸](#): When the highlight language "python" is specified explicitly, do not try to parse the code to recognize non-Python snippets.
- [#859³⁹¹⁹](#): Fix exception under certain circumstances when not finding appropriate objects to link to.

³⁹⁰³ <https://github.com/sphinx-doc/sphinx/issues/995>

³⁹⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/976>

³⁹⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/72>

³⁹⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/975>

³⁹⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/961>

³⁹⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/958>

³⁹⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/955>

³⁹¹⁰ <https://github.com/sphinx-doc/sphinx/issues/940>

³⁹¹¹ <https://github.com/sphinx-doc/sphinx/issues/920>

³⁹¹² <https://github.com/sphinx-doc/sphinx/issues/723>

³⁹¹³ <https://github.com/sphinx-doc/sphinx/issues/440>

³⁹¹⁴ <https://github.com/sphinx-doc/sphinx/issues/40>

³⁹¹⁵ <https://github.com/sphinx-doc/sphinx/issues/37>

³⁹¹⁶ <https://github.com/sphinx-doc/sphinx/issues/34>

³⁹¹⁷ <https://github.com/sphinx-doc/sphinx/issues/36>

³⁹¹⁸ <https://github.com/sphinx-doc/sphinx/issues/695>

³⁹¹⁹ <https://github.com/sphinx-doc/sphinx/issues/859>

- #860³⁹²⁰: Do not crash when encountering invalid doctest examples, just emit a warning.
- #864³⁹²¹: Fix crash with some settings of `modindex_common_prefix`.
- #862³⁹²²: Fix handling of `-D` and `-A` options on Python 3.
- #851³⁹²³: Recognize and warn about circular toctrees, instead of running into recursion errors.
- #853³⁹²⁴: Restore compatibility with Docutils trunk.
- #852³⁹²⁵: Fix HtmlHelp index entry links again.
- #854³⁹²⁶: Fix inheritance_diagram raising attribute errors on builtins.
- #832³⁹²⁷: Fix crashes when putting comments or lone terms in a glossary.
- #834³⁹²⁸, #818³⁹²⁹: Fix HTML help language/encoding mapping for all Sphinx supported languages.
- #844³⁹³⁰: Fix crashes when dealing with Unicode output in doctest extension.
- #831³⁹³¹: Provide `--project` flag in `setup_command` as advertised.
- #875³⁹³²: Fix reading config files under Python 3.
- #876³⁹³³: Fix quickstart test under Python 3.
- #870³⁹³⁴: Fix spurious KeyErrors when removing documents.
- #892³⁹³⁵: Fix single-HTML builder misbehaving with the master document in a subdirectory.
- #873³⁹³⁶: Fix assertion errors with empty only directives.
- #816³⁹³⁷: Fix encoding issues in the Qt help builder.

³⁹²⁰ <https://github.com/sphinx-doc/sphinx/issues/860>

³⁹²¹ <https://github.com/sphinx-doc/sphinx/issues/864>

³⁹²² <https://github.com/sphinx-doc/sphinx/issues/862>

³⁹²³ <https://github.com/sphinx-doc/sphinx/issues/851>

³⁹²⁴ <https://github.com/sphinx-doc/sphinx/issues/853>

³⁹²⁵ <https://github.com/sphinx-doc/sphinx/issues/852>

³⁹²⁶ <https://github.com/sphinx-doc/sphinx/issues/854>

³⁹²⁷ <https://github.com/sphinx-doc/sphinx/issues/832>

³⁹²⁸ <https://github.com/sphinx-doc/sphinx/issues/834>

³⁹²⁹ <https://github.com/sphinx-doc/sphinx/issues/818>

³⁹³⁰ <https://github.com/sphinx-doc/sphinx/issues/844>

³⁹³¹ <https://github.com/sphinx-doc/sphinx/issues/831>

³⁹³² <https://github.com/sphinx-doc/sphinx/issues/875>

³⁹³³ <https://github.com/sphinx-doc/sphinx/issues/876>

³⁹³⁴ <https://github.com/sphinx-doc/sphinx/issues/870>

³⁹³⁵ <https://github.com/sphinx-doc/sphinx/issues/892>

³⁹³⁶ <https://github.com/sphinx-doc/sphinx/issues/873>

³⁹³⁷ <https://github.com/sphinx-doc/sphinx/issues/816>

14.152 Release 1.1.2 (Nov 1, 2011) -- 1.1.1 is a silly version number anyway!

- #809³⁹³⁸: Include custom fixers in the source distribution.

14.153 Release 1.1.1 (Nov 1, 2011)

- #791³⁹³⁹: Fix QtHelp, DevHelp and HtmlHelp index entry links.
- #792³⁹⁴⁰: Include "sphinx-apidoc" in the source distribution.
- #797³⁹⁴¹: Don't crash on a misformatted glossary.
- #801³⁹⁴²: Make intersphinx work properly without SSL support.
- #805³⁹⁴³: Make the `Sphinx.add_index_to_domain` method work correctly.
- #780³⁹⁴⁴: Fix Python 2.5 compatibility.

14.154 Release 1.1 (Oct 9, 2011)

Modifiche incompatibili

- The `py:module` directive doesn't output its `platform` option value anymore. (It was the only thing that the directive did output, and therefore quite inconsistent.)
- Removed support for old dependency versions; requirements are now:
 - Pygments `>= 1.2`
 - Docutils `>= 0.7`
 - Jinja2 `>= 2.3`

³⁹³⁸ <https://github.com/sphinx-doc/sphinx/issues/809>

³⁹³⁹ <https://github.com/sphinx-doc/sphinx/issues/791>

³⁹⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/792>

³⁹⁴¹ <https://github.com/sphinx-doc/sphinx/issues/797>

³⁹⁴² <https://github.com/sphinx-doc/sphinx/issues/801>

³⁹⁴³ <https://github.com/sphinx-doc/sphinx/issues/805>

³⁹⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/780>

Funzionalità aggiunte

- Added Python 3.x support.
- New builders and subsystems:
 - Added a Texinfo builder.
 - Added i18n support for content, a `gettext` builder and related utilities.
 - Added the `websupport` library and builder.
 - [#98](#)³⁹⁴⁵: Added a `sphinx-apidoc` script that autogenerates a hierarchy of source files containing autodoc directives to document modules and packages.
 - [#273](#)³⁹⁴⁶: Add an API for adding full-text search support for languages other than English. Add support for Japanese.
- Markup:
 - [#138](#)³⁹⁴⁷: Added an `index` role, to make inline index entries.
 - [#454](#)³⁹⁴⁸: Added more index markup capabilities: marking see/seealso entries, and main entries for a given key.
 - [#460](#)³⁹⁴⁹: Allowed limiting the depth of section numbers for HTML using the `toctree`'s `numbered` option.
 - [#586](#)³⁹⁵⁰: Implemented improved `glossary` markup which allows multiple terms per definition.
 - [#478](#)³⁹⁵¹: Added `py:decorator` directive to describe decorators.
 - C++ domain now supports array definitions.
 - C++ domain now supports doc fields (`:param x:` inside directives).
 - Section headings in `only` directives are now correctly handled.
 - Added `emphasize-lines` option to source code directives.
 - [#678](#)³⁹⁵²: C++ domain now supports superclasses.
- HTML builder:
 - Added pyramid theme.
 - [#559](#)³⁹⁵³: `html_add_permalinks` is now a string giving the text to display in permalinks.

³⁹⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/98>

³⁹⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/273>

³⁹⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/138>

³⁹⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/454>

³⁹⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/460>

³⁹⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/586>

³⁹⁵¹ <https://github.com/sphinx-doc/sphinx/issues/478>

³⁹⁵² <https://github.com/sphinx-doc/sphinx/issues/678>

³⁹⁵³ <https://github.com/sphinx-doc/sphinx/issues/559>

- #259³⁹⁵⁴: HTML table rows now have even/odd CSS classes to enable "Zebra styling".
- #554³⁹⁵⁵: Add theme option `sidebarwidth` to the basic theme.
- Other builders:
 - #516³⁹⁵⁶: Added new value of the `latex_show_urls` option to show the URLs in footnotes.
 - #209³⁹⁵⁷: Added `text_newlines` and `text_sectionchars` config values.
 - Added `man_show_urls` config value.
 - #472³⁹⁵⁸: linkcheck builder: Check links in parallel, use HTTP HEAD requests and allow configuring the timeout. New config values: `linkcheck_timeout` and `linkcheck_workers`.
 - #521³⁹⁵⁹: Added `linkcheck_ignore` config value.
 - #28³⁹⁶⁰: Support row/colspans in tables in the LaTeX builder.
- Configuration and extensibility:
 - #537³⁹⁶¹: Added `nitpick_ignore`.
 - #306³⁹⁶²: Added `env-get-outdated` event.
 - `Application.add_stylesheet()` now accepts full URIs.
- Autodoc:
 - #564³⁹⁶³: Add `autodoc_docstring_signature`. When enabled (the default), autodoc retrieves the signature from the first line of the docstring, if it is found there.
 - #176³⁹⁶⁴: Provide `private-members` option for autodoc directives.
 - #520³⁹⁶⁵: Provide `special-members` option for autodoc directives.
 - #431³⁹⁶⁶: Doc comments for attributes can now be given on the same line as the assignment.
 - #437³⁹⁶⁷: autodoc now shows values of class data attributes.
 - autodoc now supports documenting the signatures of `functools.partial` objects.

³⁹⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/259>

³⁹⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/554>

³⁹⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/516>

³⁹⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/209>

³⁹⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/472>

³⁹⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/521>

³⁹⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/28>

³⁹⁶¹ <https://github.com/sphinx-doc/sphinx/issues/537>

³⁹⁶² <https://github.com/sphinx-doc/sphinx/issues/306>

³⁹⁶³ <https://github.com/sphinx-doc/sphinx/issues/564>

³⁹⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/176>

³⁹⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/520>

³⁹⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/431>

³⁹⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/437>

- Other extensions:
 - Added the `sphinx.ext.mathjax` extension.
 - #443³⁹⁶⁸: Allow referencing external graphviz files.
 - Added `inline` option to graphviz directives, and fixed the default (block-style) in LaTeX output.
 - #590³⁹⁶⁹: Added `caption` option to graphviz directives.
 - #553³⁹⁷⁰: Added `testcleanup` blocks in the doctest extension.
 - #594³⁹⁷¹: `trim_doctest_flags` now also removes `<BLANKLINE>` indicators.
 - #367³⁹⁷²: Added automatic exclusion of hidden members in inheritance diagrams, and an option to selectively enable it.
 - Added `pngmath_add_tooltips`.
 - The math extension `displaymath` directives now support `name` in addition to `label` for giving the equation label, for compatibility with Docutils.
- New locales:
 - #221³⁹⁷³: Added Swedish locale.
 - #526³⁹⁷⁴: Added Iranian locale.
 - #694³⁹⁷⁵: Added Latvian locale.
 - Added Nepali locale.
 - #714³⁹⁷⁶: Added Korean locale.
 - #766³⁹⁷⁷: Added Estonian locale.
- Bugs fixed:
 - #778³⁹⁷⁸: Fix "hide search matches" link on pages linked by search.
 - Fix the source positions referenced by the "viewcode" extension.

³⁹⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/443>

³⁹⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/590>

³⁹⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/553>

³⁹⁷¹ <https://github.com/sphinx-doc/sphinx/issues/594>

³⁹⁷² <https://github.com/sphinx-doc/sphinx/issues/367>

³⁹⁷³ <https://github.com/sphinx-doc/sphinx/issues/221>

³⁹⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/526>

³⁹⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/694>

³⁹⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/714>

³⁹⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/766>

³⁹⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/778>

14.155 Release 1.0.8 (Sep 23, 2011)

- #627³⁹⁷⁹: Fix tracebacks for AttributeErrors in autosummary generation.
- Fix the abbr role when the abbreviation has newlines in it.
- #727³⁹⁸⁰: Fix the links to search results with custom object types.
- #648³⁹⁸¹: Fix line numbers reported in warnings about undefined references.
- #696³⁹⁸², #666³⁹⁸³: Fix C++ array definitions and template arguments that are not type names.
- #633³⁹⁸⁴: Allow footnotes in section headers in LaTeX output.
- #616³⁹⁸⁵: Allow keywords to be linked via intersphinx.
- #613³⁹⁸⁶: Allow Unicode characters in production list token names.
- #720³⁹⁸⁷: Add dummy visitors for graphviz nodes for text and man.
- #704³⁹⁸⁸: Fix image file duplication bug.
- #677³⁹⁸⁹: Fix parsing of multiple signatures in C++ domain.
- #637³⁹⁹⁰: Ignore Emacs lock files when looking for source files.
- #544³⁹⁹¹: Allow .pyw extension for importable modules in autodoc.
- #700³⁹⁹²: Use \$(MAKE) in quickstart-generated Makefiles.
- #734³⁹⁹³: Make sidebar search box width consistent in browsers.
- #644³⁹⁹⁴: Fix spacing of centered figures in HTML output.
- #767³⁹⁹⁵: Safely encode SphinxError messages when printing them to sys.stderr.
- #611³⁹⁹⁶: Fix LaTeX output error with a document with no sections but a link target.
- Correctly treat built-in method descriptors as methods in autodoc.

³⁹⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/627>

³⁹⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/727>

³⁹⁸¹ <https://github.com/sphinx-doc/sphinx/issues/648>

³⁹⁸² <https://github.com/sphinx-doc/sphinx/issues/696>

³⁹⁸³ <https://github.com/sphinx-doc/sphinx/issues/666>

³⁹⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/633>

³⁹⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/616>

³⁹⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/613>

³⁹⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/720>

³⁹⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/704>

³⁹⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/677>

³⁹⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/637>

³⁹⁹¹ <https://github.com/sphinx-doc/sphinx/issues/544>

³⁹⁹² <https://github.com/sphinx-doc/sphinx/issues/700>

³⁹⁹³ <https://github.com/sphinx-doc/sphinx/issues/734>

³⁹⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/644>

³⁹⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/767>

³⁹⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/611>

- #706³⁹⁹⁷: Stop monkeypatching the Python textwrap module.
- #657³⁹⁹⁸: viewcode now works correctly with source files that have non-ASCII encoding.
- #669³⁹⁹⁹: Respect the noindex flag option in py:module directives.
- #675⁴⁰⁰⁰: Fix IndexError when including nonexistent lines with literalinclude.
- #676⁴⁰⁰¹: Respect custom function/method parameter separator strings.
- #682⁴⁰⁰²: Fix JS incompatibility with jQuery >= 1.5.
- #693⁴⁰⁰³: Fix double encoding done when writing HTMLHelp .hhk files.
- #647⁴⁰⁰⁴: Do not apply SmartyPants in parsed-literal blocks.
- C++ domain now supports array definitions.

14.156 Release 1.0.7 (Jan 15, 2011)

- #347⁴⁰⁰⁵: Fix wrong generation of directives of static methods in autosummary.
- #599⁴⁰⁰⁶: Import PIL as from PIL import Image.
- #558⁴⁰⁰⁷: Fix longtables with captions in LaTeX output.
- Make token references work as hyperlinks again in LaTeX output.
- #572⁴⁰⁰⁸: Show warnings by default when reference labels cannot be found.
- #536⁴⁰⁰⁹: Include line number when complaining about missing reference targets in nitpicky mode.
- #590⁴⁰¹⁰: Fix inline display of graphviz diagrams in LaTeX output.
- #589⁴⁰¹¹: Build using app.build() in setup command.
- Fix a bug in the inheritance diagram exception that caused base classes to be skipped if one of them is a builtin.
- Fix general index links for C++ domain objects.

³⁹⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/706>

³⁹⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/657>

³⁹⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/669>

⁴⁰⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/675>

⁴⁰⁰¹ <https://github.com/sphinx-doc/sphinx/issues/676>

⁴⁰⁰² <https://github.com/sphinx-doc/sphinx/issues/682>

⁴⁰⁰³ <https://github.com/sphinx-doc/sphinx/issues/693>

⁴⁰⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/647>

⁴⁰⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/347>

⁴⁰⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/599>

⁴⁰⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/558>

⁴⁰⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/572>

⁴⁰⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/536>

⁴⁰¹⁰ <https://github.com/sphinx-doc/sphinx/issues/590>

⁴⁰¹¹ <https://github.com/sphinx-doc/sphinx/issues/589>

- [#332](#)⁴⁰¹²: Make admonition boundaries in LaTeX output visible.
- [#573](#)⁴⁰¹³: Fix KeyErrors occurring on rebuild after removing a file.
- Fix a traceback when removing files with globbed toctrees.
- If an autodoc object cannot be imported, always re-read the document containing the directive on next build.
- If an autodoc object cannot be imported, show the full traceback of the import error.
- Fix a bug where the removal of download files and images wasn't noticed.
- [#571](#)⁴⁰¹⁴: Implement ~ cross-reference prefix for the C domain.
- Fix regression of LaTeX output with the fix of [#556](#)⁴⁰¹⁵.
- [#568](#)⁴⁰¹⁶: Fix lookup of class attribute documentation on descriptors so that comment documentation now works.
- Fix traceback with `only` directives preceded by targets.
- Fix tracebacks occurring for duplicate C++ domain objects.
- Fix JavaScript domain links to objects with \$ in their name.

14.157 Release 1.0.6 (Jan 04, 2011)

- [#581](#)⁴⁰¹⁷: Fix traceback in Python domain for empty cross-reference targets.
- [#283](#)⁴⁰¹⁸: Fix literal block display issues on Chrome browsers.
- [#383](#)⁴⁰¹⁹, [#148](#)⁴⁰²⁰: Support sorting a limited range of accented characters in the general index and the glossary.
- [#570](#)⁴⁰²¹: Try decoding `-D` and `-A` command-line arguments with the locale's preferred encoding.
- [#528](#)⁴⁰²²: Observe `locale_dirs` when looking for the JS translations file.
- [#574](#)⁴⁰²³: Add special code for better support of Japanese documents in the LaTeX builder.

⁴⁰¹² <https://github.com/sphinx-doc/sphinx/issues/332>

⁴⁰¹³ <https://github.com/sphinx-doc/sphinx/issues/573>

⁴⁰¹⁴ <https://github.com/sphinx-doc/sphinx/issues/571>

⁴⁰¹⁵ <https://github.com/sphinx-doc/sphinx/issues/556>

⁴⁰¹⁶ <https://github.com/sphinx-doc/sphinx/issues/568>

⁴⁰¹⁷ <https://github.com/sphinx-doc/sphinx/issues/581>

⁴⁰¹⁸ <https://github.com/sphinx-doc/sphinx/issues/283>

⁴⁰¹⁹ <https://github.com/sphinx-doc/sphinx/issues/383>

⁴⁰²⁰ <https://github.com/sphinx-doc/sphinx/issues/148>

⁴⁰²¹ <https://github.com/sphinx-doc/sphinx/issues/570>

⁴⁰²² <https://github.com/sphinx-doc/sphinx/issues/528>

⁴⁰²³ <https://github.com/sphinx-doc/sphinx/issues/574>

- Regression of #77⁴⁰²⁴: If there is only one parameter given with `:param:` markup, the bullet list is now suppressed again.
- #556⁴⁰²⁵: Fix missing paragraph breaks in LaTeX output in certain situations.
- #567⁴⁰²⁶: Emit the `autodoc-process-docstring` event even for objects without a docstring so that it can add content.
- #565⁴⁰²⁷: In the LaTeX builder, not only literal blocks require different table handling, but also quite a few other list-like block elements.
- #515⁴⁰²⁸: Fix tracebacks in the viewcode extension for Python objects that do not have a valid signature.
- Fix strange reports of line numbers for warnings generated from autodoc-included docstrings, due to different behavior depending on Docutils version.
- Several fixes to the C++ domain.

14.158 Release 1.0.5 (Nov 12, 2010)

- #557⁴⁰²⁹: Add CSS styles required by Docutils 0.7 for aligned images and figures.
- In the Makefile generated by LaTeX output, do not delete pdf files on clean; they might be required images.
- #535⁴⁰³⁰: Fix LaTeX output generated for line blocks.
- #544⁴⁰³¹: Allow `.pyw` as a source file extension.

14.159 Release 1.0.4 (Sep 17, 2010)

- #524⁴⁰³²: Open intersphinx inventories in binary mode on Windows, since version 2 contains zlib-compressed data.
- #513⁴⁰³³: Allow giving non-local URIs for JavaScript files, e.g. in the JSMath extension.
- #512⁴⁰³⁴: Fix traceback when `intersphinx_mapping` is empty.

⁴⁰²⁴ <https://github.com/sphinx-doc/sphinx/issues/77>

⁴⁰²⁵ <https://github.com/sphinx-doc/sphinx/issues/556>

⁴⁰²⁶ <https://github.com/sphinx-doc/sphinx/issues/567>

⁴⁰²⁷ <https://github.com/sphinx-doc/sphinx/issues/565>

⁴⁰²⁸ <https://github.com/sphinx-doc/sphinx/issues/515>

⁴⁰²⁹ <https://github.com/sphinx-doc/sphinx/issues/557>

⁴⁰³⁰ <https://github.com/sphinx-doc/sphinx/issues/535>

⁴⁰³¹ <https://github.com/sphinx-doc/sphinx/issues/544>

⁴⁰³² <https://github.com/sphinx-doc/sphinx/issues/524>

⁴⁰³³ <https://github.com/sphinx-doc/sphinx/issues/513>

⁴⁰³⁴ <https://github.com/sphinx-doc/sphinx/issues/512>

14.160 Release 1.0.3 (Aug 23, 2010)

- [#495](#)⁴⁰³⁵: Fix internal vs. external link distinction for links coming from a Docutils table-of-contents.
- [#494](#)⁴⁰³⁶: Fix the maxdepth option for the toctree() template callable when used with collapse=True.
- [#507](#)⁴⁰³⁷: Fix crash parsing Python argument lists containing brackets in string literals.
- [#501](#)⁴⁰³⁸: Fix regression when building LaTeX docs with figures that don't have captions.
- [#510](#)⁴⁰³⁹: Fix inheritance diagrams for classes that are not picklable.
- [#497](#)⁴⁰⁴⁰: Introduce separate background color for the sidebar collapse button, making it easier to see.
- [#502](#)⁴⁰⁴¹, [#503](#)⁴⁰⁴², [#496](#)⁴⁰⁴³: Fix small layout bugs in several builtin themes.

14.161 Release 1.0.2 (Aug 14, 2010)

- [#490](#)⁴⁰⁴⁴: Fix cross-references to objects of types added by the `add_object_type()` API function.
- Fix handling of doc field types for different directive types.
- Allow breaking long signatures, continuing with backlash-escaped newlines.
- Fix unwanted styling of C domain references (because of a namespace clash with Pygments styles).
- Allow references to PEPs and RFCs with explicit anchors.
- [#471](#)⁴⁰⁴⁵: Fix LaTeX references to figures.
- [#482](#)⁴⁰⁴⁶: When doing a non-exact search, match only the given type of object.
- [#481](#)⁴⁰⁴⁷: Apply non-exact search for Python reference targets with `.name` for modules too.

⁴⁰³⁵ <https://github.com/sphinx-doc/sphinx/issues/495>

⁴⁰³⁶ <https://github.com/sphinx-doc/sphinx/issues/494>

⁴⁰³⁷ <https://github.com/sphinx-doc/sphinx/issues/507>

⁴⁰³⁸ <https://github.com/sphinx-doc/sphinx/issues/501>

⁴⁰³⁹ <https://github.com/sphinx-doc/sphinx/issues/510>

⁴⁰⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/497>

⁴⁰⁴¹ <https://github.com/sphinx-doc/sphinx/issues/502>

⁴⁰⁴² <https://github.com/sphinx-doc/sphinx/issues/503>

⁴⁰⁴³ <https://github.com/sphinx-doc/sphinx/issues/496>

⁴⁰⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/490>

⁴⁰⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/471>

⁴⁰⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/482>

⁴⁰⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/481>

- #484⁴⁰⁴⁸: Fix crash when duplicating a parameter in an info field list.
- #487⁴⁰⁴⁹: Fix setting the default role to one provided by the oldcmakup extension.
- #488⁴⁰⁵⁰: Fix crash when json-py is installed, which provides a json module but is incompatible to simplejson.
- #480⁴⁰⁵¹: Fix handling of target naming in intersphinx.
- #486⁴⁰⁵²: Fix removal of ! for all cross-reference roles.

14.162 Release 1.0.1 (Jul 27, 2010)

- #470⁴⁰⁵³: Fix generated target names for reST domain objects; they are not in the same namespace.
- #266⁴⁰⁵⁴: Add Bengali language.
- #473⁴⁰⁵⁵: Fix a bug in parsing JavaScript object names.
- #474⁴⁰⁵⁶: Fix building with SingleHTMLBuilder when there is no toctree.
- Fix display names for objects linked to by intersphinx with explicit targets.
- Fix building with the JSON builder.
- Fix hyperrefs in object descriptions for LaTeX.

14.163 Release 1.0 (Jul 23, 2010)

Modifiche incompatibili

- Support for domains has been added. A domain is a collection of directives and roles that all describe objects belonging together, e.g. elements of a programming language. A few builtin domains are provided:
 - Python
 - C
 - C++
 - JavaScript

⁴⁰⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/484>

⁴⁰⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/487>

⁴⁰⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/488>

⁴⁰⁵¹ <https://github.com/sphinx-doc/sphinx/issues/480>

⁴⁰⁵² <https://github.com/sphinx-doc/sphinx/issues/486>

⁴⁰⁵³ <https://github.com/sphinx-doc/sphinx/issues/470>

⁴⁰⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/266>

⁴⁰⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/473>

⁴⁰⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/474>

- `reStructuredText`
- The old markup for defining and linking to C directives is now deprecated. It will not work anymore in future versions without activating the `oldcm Markup` extension; in Sphinx 1.0, it is activated by default.
- Removed support for old dependency versions; requirements are now:
 - Docutils `>= 0.5`
 - Jinja2 `>= 2.2`
- Removed deprecated elements:
 - `exclude_dirs` config value
 - `sphinx.builder` module

Funzionalità aggiunte

- General:
 - Added a "nitpicky" mode that emits warnings for all missing references. It is activated by the `sphinx-build -n` command-line switch or the `nitpicky` config value.
 - Added `latexpdf` target in quickstart Makefile.
- Markup:
 - The `menuselection` and `guilabel` roles now support ampersand accelerators.
 - New more compact doc field syntax is now recognized: `:param type name: description`.
 - Added `tab-width` option to `literalinclude` directive.
 - Added `titlesonly` option to `toctree` directive.
 - Added the `prepend` and `append` options to the `literalinclude` directive.
 - [#284](https://github.com/sphinx-doc/sphinx/issues/284)⁴⁰⁵⁷: All docinfo metadata is now put into the document metadata, not just the author.
 - The `ref` role can now also reference tables by caption.
 - The `include`⁴⁰⁵⁸ directive now supports absolute paths, which are interpreted as relative to the source directory.
 - In the Python domain, references like `:func:`.name`` now look for matching names with any prefix if no direct match is found.
- Configuration:
 - Added `rst_prolog` config value.

⁴⁰⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/284>

⁴⁰⁵⁸ <https://docutils.sourceforge.io/docs/ref/rst/directives.html#include>

- Added `html_secnumber_suffix` config value to control section numbering format.
 - Added `html_compact_lists` config value to control Docutils' compact lists feature.
 - The `html_sidebars` config value can now contain patterns as keys, and the values can be lists that explicitly select which sidebar templates should be rendered. That means that the builtin sidebar contents can be included only selectively.
 - `html_static_path` can now contain single file entries.
 - The new universal config value `exclude_patterns` makes the old `unused_docs`, `exclude_trees` and `exclude_dirnames` obsolete.
 - Added `html_output_encoding` config value.
 - Added the `latex_docclass` config value and made the "twoside" documentclass option overridable by "oneside".
 - Added the `trim_doctest_flags` config value, which is true by default.
 - Added `html_show_copyright` config value.
 - Added `latex_show_pagerefs`` and `latex_show_urls` config values.
 - The behavior of `html_file_suffix`` changed slightly: the empty string now means "no suffix" instead of "default suffix", use `None` for "default suffix".
- New builders:
 - Added a builder for the Epub format.
 - Added a builder for manual pages.
 - Added a single-file HTML builder.
 - HTML output:
 - Inline roles now get a CSS class with their name, allowing styles to customize their appearance. Domain-specific roles get two classes, `domain` and `domain-rolename`.
 - References now get the class `internal` if they are internal to the whole project, as opposed to internal to the current page.
 - External references can be styled differently with the new `externalrefs` theme option for the default theme.
 - In the default theme, the sidebar can experimentally now be made collapsible using the new `collapsiblesidebar` theme option.
 - [#129](https://github.com/sphinx-doc/sphinx/issues/129)⁴⁰⁵⁹: Toctrees are now wrapped in a `div` tag with class `toctree-wrapper` in HTML output.
 - The `toctree` callable in templates now has a `maxdepth` keyword argument to control the depth of the generated tree.

⁴⁰⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/129>

- The `toctree` callable in templates now accepts a `titles_only` keyword argument.
- Added `htmltitle` block in layout template.
- In the JavaScript search, allow searching for object names including the module name, like `sys.argv`.
- Added new theme `haiku`, inspired by the Haiku OS user guide.
- Added new theme `nature`.
- Added new theme `agogo`, created by Andi Albrecht.
- Added new theme `scrolls`, created by Armin Ronacher.
- [#193](https://github.com/sphinx-doc/sphinx/issues/193)⁴⁰⁶⁰: Added a `visitedlinkcolor` theme option to the default theme.
- [#322](https://github.com/sphinx-doc/sphinx/issues/322)⁴⁰⁶¹: Improved responsiveness of the search page by loading the search index asynchronously.
- Extension API:
 - Added `html-collect-pages`.
 - Added `needs_sphinx` config value and `require_sphinx()` application API method.
 - [#200](https://github.com/sphinx-doc/sphinx/issues/200)⁴⁰⁶²: Added `add_stylesheet()` application API method.
- Extensions:
 - Added the `viewcode` extension.
 - Added the `extlinks` extension.
 - Added support for source ordering of members in autodoc, with `autodoc_member_order = 'bysource'`.
 - Added `autodoc_default_flags` config value, which can be used to select default flags for all autodoc directives.
 - Added a way for intersphinx to refer to named labels in other projects, and to specify the project you want to link to.
 - [#280](https://github.com/sphinx-doc/sphinx/issues/280)⁴⁰⁶³: Autodoc can now document instance attributes assigned in `__init__` methods.
 - Many improvements and fixes to the `autosummary` extension, thanks to Pauli Virtanen.
 - [#309](https://github.com/sphinx-doc/sphinx/issues/309)⁴⁰⁶⁴: The `graphviz` extension can now output SVG instead of PNG images, controlled by the `graphviz_output_format` config value.
 - Added `alt` option to `graphviz` extension directives.

⁴⁰⁶⁰ <https://github.com/sphinx-doc/sphinx/issues/193>

⁴⁰⁶¹ <https://github.com/sphinx-doc/sphinx/issues/322>

⁴⁰⁶² <https://github.com/sphinx-doc/sphinx/issues/200>

⁴⁰⁶³ <https://github.com/sphinx-doc/sphinx/issues/280>

⁴⁰⁶⁴ <https://github.com/sphinx-doc/sphinx/issues/309>

- Added `exclude` argument to `autodoc.between()`.
- Translations:
 - Added Croatian translation, thanks to Bojan Mihelač.
 - Added Turkish translation, thanks to Firat Ozgul.
 - Added Catalan translation, thanks to Pau Fernández.
 - Added simplified Chinese translation.
 - Added Danish translation, thanks to Hjorth Larsen.
 - Added Lithuanian translation, thanks to Dalius Dobravolskas.
- Bugs fixed:
 - [#445](#)⁴⁰⁶⁵: Fix links to result pages when using the search function of HTML built with the `dirhtml` builder.
 - [#444](#)⁴⁰⁶⁶: In templates, properly re-escape values treated with the "striptags" Jinja filter.

14.164 Release 0.6.7 (Jun 05, 2010)

- [#440](#)⁴⁰⁶⁷: Remove usage of a Python `>= 2.5` API in the `literalinclude` directive.
- Fix a bug that prevented some references being generated in the LaTeX builder.
- [#428](#)⁴⁰⁶⁸: Add some missing CSS styles for standard Docutils classes.
- [#432](#)⁴⁰⁶⁹: Fix UnicodeErrors while building LaTeX in translated locale.

14.165 Release 0.6.6 (May 25, 2010)

- Handle raw nodes in the text writer.
- Fix a problem the Qt help project generated by the `qthelp` builder that would lead to no content being displayed in the Qt Assistant.
- [#393](#)⁴⁰⁷⁰: Fix the usage of Unicode characters in mathematic formulas when using the `pngmath` extension.
- [#404](#)⁴⁰⁷¹: Make `\and` work properly in the author field of the `latex_documents` setting.

⁴⁰⁶⁵ <https://github.com/sphinx-doc/sphinx/issues/445>

⁴⁰⁶⁶ <https://github.com/sphinx-doc/sphinx/issues/444>

⁴⁰⁶⁷ <https://github.com/sphinx-doc/sphinx/issues/440>

⁴⁰⁶⁸ <https://github.com/sphinx-doc/sphinx/issues/428>

⁴⁰⁶⁹ <https://github.com/sphinx-doc/sphinx/issues/432>

⁴⁰⁷⁰ <https://github.com/sphinx-doc/sphinx/issues/393>

⁴⁰⁷¹ <https://github.com/sphinx-doc/sphinx/issues/404>

- [#409](#)⁴⁰⁷²: Make the `highlight_language` config value work properly in the LaTeX builder.
- [#418](#)⁴⁰⁷³: Allow relocation of the translation JavaScript files to the system directory on Unix systems.
- [#414](#)⁴⁰⁷⁴: Fix handling of Windows newlines in files included with the `literalinclude` directive.
- [#377](#)⁴⁰⁷⁵: Fix crash in linkcheck builder.
- [#387](#)⁴⁰⁷⁶: Fix the display of search results in `dirhtml` output.
- [#376](#)⁴⁰⁷⁷: In autodoc, fix display of parameter defaults containing backslashes.
- [#370](#)⁴⁰⁷⁸: Fix handling of complex list item labels in LaTeX output.
- [#374](#)⁴⁰⁷⁹: Make the `doctest_path` config value of the doctest extension actually work.
- Fix the handling of multiple toctrees when creating the global TOC for the `toctree()` template function.
- Fix the handling of hidden toctrees when creating the global TOC for the `toctree()` template function.
- Fix the handling of nested lists in the text writer.
- [#362](#)⁴⁰⁸⁰: In autodoc, check for the existence of `__self__` on function objects before accessing it.
- [#353](#)⁴⁰⁸¹: Strip leading and trailing whitespace when extracting search words in the search function.

14.166 Release 0.6.5 (Mar 01, 2010)

- In autodoc, fix the omission of some module members explicitly documented using documentation comments.
- [#345](#)⁴⁰⁸²: Fix cropping of sidebar scroll bar with `stickysidebar` option of the default theme.
- [#341](#)⁴⁰⁸³: Always generate UNIX newlines in the quickstart Makefile.

⁴⁰⁷² <https://github.com/sphinx-doc/sphinx/issues/409>

⁴⁰⁷³ <https://github.com/sphinx-doc/sphinx/issues/418>

⁴⁰⁷⁴ <https://github.com/sphinx-doc/sphinx/issues/414>

⁴⁰⁷⁵ <https://github.com/sphinx-doc/sphinx/issues/377>

⁴⁰⁷⁶ <https://github.com/sphinx-doc/sphinx/issues/387>

⁴⁰⁷⁷ <https://github.com/sphinx-doc/sphinx/issues/376>

⁴⁰⁷⁸ <https://github.com/sphinx-doc/sphinx/issues/370>

⁴⁰⁷⁹ <https://github.com/sphinx-doc/sphinx/issues/374>

⁴⁰⁸⁰ <https://github.com/sphinx-doc/sphinx/issues/362>

⁴⁰⁸¹ <https://github.com/sphinx-doc/sphinx/issues/353>

⁴⁰⁸² <https://github.com/sphinx-doc/sphinx/issues/345>

⁴⁰⁸³ <https://github.com/sphinx-doc/sphinx/issues/341>

- [#338](#)⁴⁰⁸⁴: Fix running with `-C` under Windows.
- In autodoc, allow customizing the signature of an object where the built-in mechanism fails.
- [#331](#)⁴⁰⁸⁵: Fix output for enumerated lists with start values in LaTeX.
- Make the `start-after` and `end-before` options to the `literalinclude` directive work correctly if not used together.
- [#321](#)⁴⁰⁸⁶: Fix link generation in the LaTeX builder.

14.167 Release 0.6.4 (Jan 12, 2010)

- Improve the handling of non-Unicode strings in the configuration.
- [#316](#)⁴⁰⁸⁷: Catch `OSErrors` occurring when calling `graphviz` with arguments it doesn't understand.
- Restore compatibility with `Pygments` `>= 1.2`.
- [#295](#)⁴⁰⁸⁸: Fix escaping of `hyperref` targets in LaTeX output.
- [#302](#)⁴⁰⁸⁹: Fix links generated by the `:doc:` role for LaTeX output.
- [#286](#)⁴⁰⁹⁰: collect todo nodes after the whole document has been read; this allows placing substitution references in todo items.
- [#294](#)⁴⁰⁹¹: do not ignore an explicit `today` config value in a LaTeX build.
- The `alt` text of inheritance diagrams is now much cleaner.
- Ignore images in section titles when generating link captions.
- [#310](#)⁴⁰⁹²: support exception messages in the `testoutput` blocks of the `doctest` extension.
- [#293](#)⁴⁰⁹³: line blocks are styled properly in HTML output.
- [#285](#)⁴⁰⁹⁴: make the `locale_dirs` config value work again.
- [#303](#)⁴⁰⁹⁵: `html_context` values given on the command line via `-A` should not override other values given in `conf.py`.

⁴⁰⁸⁴ <https://github.com/sphinx-doc/sphinx/issues/338>

⁴⁰⁸⁵ <https://github.com/sphinx-doc/sphinx/issues/331>

⁴⁰⁸⁶ <https://github.com/sphinx-doc/sphinx/issues/321>

⁴⁰⁸⁷ <https://github.com/sphinx-doc/sphinx/issues/316>

⁴⁰⁸⁸ <https://github.com/sphinx-doc/sphinx/issues/295>

⁴⁰⁸⁹ <https://github.com/sphinx-doc/sphinx/issues/302>

⁴⁰⁹⁰ <https://github.com/sphinx-doc/sphinx/issues/286>

⁴⁰⁹¹ <https://github.com/sphinx-doc/sphinx/issues/294>

⁴⁰⁹² <https://github.com/sphinx-doc/sphinx/issues/310>

⁴⁰⁹³ <https://github.com/sphinx-doc/sphinx/issues/293>

⁴⁰⁹⁴ <https://github.com/sphinx-doc/sphinx/issues/285>

⁴⁰⁹⁵ <https://github.com/sphinx-doc/sphinx/issues/303>

- Fix a bug preventing incremental rebuilds for the `dirhtml` builder.
- [#299⁴⁰⁹⁶](#): Fix the mangling of quotes in some literal blocks.
- [#292⁴⁰⁹⁷](#): Fix path to the search index for the `dirhtml` builder.
- Fix a Jython compatibility issue: make the dependence on the `parser` module optional.
- [#238⁴⁰⁹⁸](#): In autodoc, catch all errors that occur on module import, not just `ImportError`.
- Fix the handling of non-data, but non-method descriptors in autodoc.
- When copying file times, ignore `OSErrors` raised by `os.utime()`.

14.168 Release 0.6.3 (Sep 03, 2009)

- Properly add C module filenames as dependencies in autodoc.
- [#253⁴⁰⁹⁹](#): Ignore `graphviz` directives without content instead of raising an unhandled exception.
- [#241⁴¹⁰⁰](#): Fix a crash building LaTeX output for documents that contain a `todolist` directive.
- [#252⁴¹⁰¹](#): Make it easier to change the build dir in the Makefiles generated by quickstart.
- [#220⁴¹⁰²](#): Fix CSS so that `displaymath` really is centered.
- [#222⁴¹⁰³](#): Allow the "Footnotes" header to be translated.
- [#225⁴¹⁰⁴](#): Don't add whitespace in generated HTML after inline tags.
- [#227⁴¹⁰⁵](#): Make `literalinclude` work when the document's path name contains non-ASCII characters.
- [#229⁴¹⁰⁶](#): Fix autodoc failures with members that raise errors on `getattr()`.
- [#205⁴¹⁰⁷](#): When copying files, don't copy full stat info, only modification times.
- [#232⁴¹⁰⁸](#): Support non-ASCII metadata in Qt help builder.
- Properly format bullet lists nested in definition lists for LaTeX.
- Section titles are now allowed inside `only` directives.

⁴⁰⁹⁶ <https://github.com/sphinx-doc/sphinx/issues/299>

⁴⁰⁹⁷ <https://github.com/sphinx-doc/sphinx/issues/292>

⁴⁰⁹⁸ <https://github.com/sphinx-doc/sphinx/issues/238>

⁴⁰⁹⁹ <https://github.com/sphinx-doc/sphinx/issues/253>

⁴¹⁰⁰ <https://github.com/sphinx-doc/sphinx/issues/241>

⁴¹⁰¹ <https://github.com/sphinx-doc/sphinx/issues/252>

⁴¹⁰² <https://github.com/sphinx-doc/sphinx/issues/220>

⁴¹⁰³ <https://github.com/sphinx-doc/sphinx/issues/222>

⁴¹⁰⁴ <https://github.com/sphinx-doc/sphinx/issues/225>

⁴¹⁰⁵ <https://github.com/sphinx-doc/sphinx/issues/227>

⁴¹⁰⁶ <https://github.com/sphinx-doc/sphinx/issues/229>

⁴¹⁰⁷ <https://github.com/sphinx-doc/sphinx/issues/205>

⁴¹⁰⁸ <https://github.com/sphinx-doc/sphinx/issues/232>

- #201⁴¹⁰⁹: Make centered directive work in LaTeX output.
- #206⁴¹¹⁰: Refuse to overwrite an existing master document in sphinx-quickstart.
- #208⁴¹¹¹: Use MS-sanctioned locale settings, determined by the language config option, in the HTML help builder.
- #210⁴¹¹²: Fix nesting of HTML tags for displayed math from pngmath extension.
- #213⁴¹¹³: Fix centering of images in LaTeX output.
- #211⁴¹¹⁴: Fix compatibility with Docutils 0.5.

14.169 Release 0.6.2 (Jun 16, 2009)

- #130⁴¹¹⁵: Fix obscure IndexError in doctest extension.
- #167⁴¹¹⁶: Make glossary sorting case-independent.
- #196⁴¹¹⁷: Add a warning if an extension module doesn't have a `setup()` function.
- #158⁴¹¹⁸: Allow `'..'` in template names, and absolute template paths; Jinja 2 by default disables both.
- When highlighting Python code, ignore extra indentation before trying to parse it as Python.
- #191⁴¹¹⁹: Don't escape the tilde in URIs in LaTeX.
- Don't consider contents of source comments for the search index.
- Set the default encoding to `utf-8-sig` to handle files with a UTF-8 BOM correctly.
- #178⁴¹²⁰: apply `add_function_parentheses` config value to C functions as promised.
- #173⁴¹²¹: Respect the Docutils `title` directive.
- #172⁴¹²²: The `obj` role now links to modules as promised.
- #19⁴¹²³: Tables now can have a "longtable" class, in order to get correctly broken into pages in LaTeX output.

⁴¹⁰⁹ <https://github.com/sphinx-doc/sphinx/issues/201>

⁴¹¹⁰ <https://github.com/sphinx-doc/sphinx/issues/206>

⁴¹¹¹ <https://github.com/sphinx-doc/sphinx/issues/208>

⁴¹¹² <https://github.com/sphinx-doc/sphinx/issues/210>

⁴¹¹³ <https://github.com/sphinx-doc/sphinx/issues/213>

⁴¹¹⁴ <https://github.com/sphinx-doc/sphinx/issues/211>

⁴¹¹⁵ <https://github.com/sphinx-doc/sphinx/issues/130>

⁴¹¹⁶ <https://github.com/sphinx-doc/sphinx/issues/167>

⁴¹¹⁷ <https://github.com/sphinx-doc/sphinx/issues/196>

⁴¹¹⁸ <https://github.com/sphinx-doc/sphinx/issues/158>

⁴¹¹⁹ <https://github.com/sphinx-doc/sphinx/issues/191>

⁴¹²⁰ <https://github.com/sphinx-doc/sphinx/issues/178>

⁴¹²¹ <https://github.com/sphinx-doc/sphinx/issues/173>

⁴¹²² <https://github.com/sphinx-doc/sphinx/issues/172>

⁴¹²³ <https://github.com/sphinx-doc/sphinx/issues/19>

- Look for Sphinx message catalogs in the system default path before trying `sphinx/locale`.
- Fix the search for methods via `"classname.methodname"`.
- [#155](#)⁴¹²⁴: Fix Python 2.4 compatibility: exceptions are old-style classes there.
- [#150](#)⁴¹²⁵: Fix display of the "sphinxdoc" theme on Internet Explorer versions 6 and 7.
- [#146](#)⁴¹²⁶: Don't fail to generate LaTeX when the user has an active `.docutils` configuration.
- [#29](#)⁴¹²⁷: Don't generate visible `"-{-}"` in option lists in LaTeX.
- Fix cross-reference roles when put into substitutions.
- Don't put image "alt" text into table-of-contents entries.
- In the LaTeX writer, do not raise an exception on too many section levels, just use the "subparagraph" level for all of them.
- [#145](#)⁴¹²⁸: Fix autodoc problem with automatic members that refuse to be `getattr()`'d from their parent.
- If specific filenames to build are given on the command line, check that they are within the source directory.
- Fix autodoc crash for objects without a `__name__`.
- Fix intersphinx for installations without `urllib2.HTTPSHandler`.
- [#134](#)⁴¹²⁹: Fix pending_xref leftover nodes when using the `todo` directive from the `todo` extension.

14.170 Release 0.6.1 (Mar 26, 2009)

- [#135](#)⁴¹³⁰: Fix problems with LaTeX output and the graphviz extension.
- [#132](#)⁴¹³¹: Include the autosummary "module" template in the distribution.

⁴¹²⁴ <https://github.com/sphinx-doc/sphinx/issues/155>

⁴¹²⁵ <https://github.com/sphinx-doc/sphinx/issues/150>

⁴¹²⁶ <https://github.com/sphinx-doc/sphinx/issues/146>

⁴¹²⁷ <https://github.com/sphinx-doc/sphinx/issues/29>

⁴¹²⁸ <https://github.com/sphinx-doc/sphinx/issues/145>

⁴¹²⁹ <https://github.com/sphinx-doc/sphinx/issues/134>

⁴¹³⁰ <https://github.com/sphinx-doc/sphinx/issues/135>

⁴¹³¹ <https://github.com/sphinx-doc/sphinx/issues/132>

14.171 Release 0.6 (Mar 24, 2009)

New features added

- Incompatible changes:
 - Templating now requires the Jinja2 library, which is an enhanced version of the old Jinja1 engine. Since the syntax and semantic is largely the same, very few fixes should be necessary in custom templates.
 - The "document" div tag has been moved out of the `layout.html` template's "document" block, because the closing tag was already outside. If you overwrite this block, you need to remove your "document" div tag as well.
 - The `autodoc_skip_member` event now also gets to decide whether to skip members whose name starts with underscores. Previously, these members were always automatically skipped. Therefore, if you handle this event, add something like this to your event handler to restore the old behavior:

```
if name.startswith('_ _'):
    return True
```

- Theming support, see the new section in the documentation.
- Markup:
 - Due to popular demand, added a `:doc:` role which directly links to another document without the need of creating a label to which a `:ref:` could link to.
 - [#4](https://github.com/sphinx-doc/sphinx/issues/4)⁴¹³²: Added a `:download:` role that marks a non-document file for inclusion into the HTML output and links to it.
 - Added an `only` directive that can selectively include text based on enabled "tags". Tags can be given on the command line. Also, the current builder output format (e.g. "html" or "latex") is always a defined tag.
 - [#10](https://github.com/sphinx-doc/sphinx/issues/10)⁴¹³³: Added HTML section numbers, enabled by giving a `:numbered:` flag to the `toctree` directive.
 - [#114](https://github.com/sphinx-doc/sphinx/issues/114)⁴¹³⁴: Added an `abbr` role to markup abbreviations and acronyms.
 - The `literalinclude` directive now supports several more options, to include only parts of a file.
 - The `toctree` directive now supports a `:hidden:` flag, which will prevent links from being generated in place of the directive -- this allows you to define your document structure, but place the links yourself.

⁴¹³² <https://github.com/sphinx-doc/sphinx/issues/4>

⁴¹³³ <https://github.com/sphinx-doc/sphinx/issues/10>

⁴¹³⁴ <https://github.com/sphinx-doc/sphinx/issues/114>

- [#123](https://github.com/sphinx-doc/sphinx/issues/123)⁴¹³⁵: The `glossary` directive now supports a `:sorted:` flag that sorts glossary entries alphabetically.
 - Paths to images, literal include files and download files can now be absolute (like `/images/foo.png`). They are treated as relative to the top source directory.
 - [#52](https://github.com/sphinx-doc/sphinx/issues/52)⁴¹³⁶: There is now a `hlist` directive, creating a compact list by placing distributing items into multiple columns.
 - [#77](https://github.com/sphinx-doc/sphinx/issues/77)⁴¹³⁷: If a description environment with info field list only contains one `:param:` entry, no bullet list is generated.
 - [#6](https://github.com/sphinx-doc/sphinx/issues/6)⁴¹³⁸: Don't generate redundant `` for top-level TOC tree items, which leads to a visual separation of TOC entries.
 - [#23](https://github.com/sphinx-doc/sphinx/issues/23)⁴¹³⁹: Added a `classmethod` directive along with `method` and `staticmethod`.
 - Scaled images now get a link to the unscaled version.
 - SVG images are now supported in HTML (via `<object>` and `<embed>` tags).
 - Added a `toctree` callable to the templates, and the ability to include external links in toctrees. The 'collapse' keyword argument indicates whether or not to only display subitems of the current page. (Defaults to True.)
- Configuration:
 - The new config value `rst_epilog` can contain reST that is appended to each source file that is read. This is the right place for global substitutions.
 - The new `html_add_permalink` config value can be used to switch off the generated "paragraph sign" permalinks for each heading and definition environment.
 - The new `html_show_sourcelink` config value can be used to switch off the links to the reST sources in the sidebar.
 - The default value for `htmlhelp_basename` is now the project title, cleaned up as a filename.
 - The new `modindex_common_prefix` config value can be used to ignore certain package names for module index sorting.
 - The new `trim_footnote_reference_space` config value mirrors the Docutils config value of the same name and removes the space before a footnote reference that is necessary for reST to recognize the reference.
 - The new `latex_additional_files` config value can be used to copy files (that Sphinx doesn't copy automatically, e.g. if they are referenced in custom LaTeX added in `latex_elements`) to the build directory.

- Builders:

⁴¹³⁵ <https://github.com/sphinx-doc/sphinx/issues/123>

⁴¹³⁶ <https://github.com/sphinx-doc/sphinx/issues/52>

⁴¹³⁷ <https://github.com/sphinx-doc/sphinx/issues/77>

⁴¹³⁸ <https://github.com/sphinx-doc/sphinx/issues/6>

⁴¹³⁹ <https://github.com/sphinx-doc/sphinx/issues/23>

- The HTML builder now stores a small file named `.buildinfo` in its output directory. It stores a hash of config values that can be used to determine if a full rebuild needs to be done (e.g. after changing `html_theme`).
- New builder for Qt help collections, by Antonio Valentino.
- The new `DirectoryHTMLBuilder` (short name `dirhtml`) creates a separate directory for every page, and places the page there in a file called `index.html`. Therefore, page URLs and links don't need to contain `.html`.
- The new `html_link_suffix` config value can be used to select the suffix of generated links between HTML files.
- [#96](https://github.com/sphinx-doc/sphinx/issues/96)⁴¹⁴⁰: The LaTeX builder now supports figures wrapped by text, when using the `figwidth` option and right/left alignment.
- New translations:
 - Italian by Sandro Dentella.
 - Ukrainian by Petro Sasnyk.
 - Finnish by Jukka Inkeri.
 - Russian by Alexander Smishlajev.
- Extensions and API:
 - New `graphviz` extension to embed graphviz graphs.
 - New `inheritance_diagram` extension to embed... inheritance diagrams!
 - New `autosummary` extension that generates summaries of modules and automatic documentation of modules.
 - Autodoc now has a reusable Python API, which can be used to create custom types of objects to auto-document (e.g. Zope interfaces). See also `Sphinx.add_autodocumenter()`.
 - Autodoc now handles documented attributes.
 - Autodoc now handles inner classes and their methods.
 - Autodoc can document classes as functions now if explicitly marked with `autofunction`.
 - Autodoc can now exclude single members from documentation via the `exclude-members` option.
 - Autodoc can now order members either alphabetically (like previously) or by member type; configurable either with the config value `autodoc_member_order` or a `member-order` option per directive.
 - The function `Sphinx.add_directive()` now also supports Docutils 0.5-style directive classes. If they inherit from `sphinx.util.compat.Directive`, they also work with Docutils 0.4.

⁴¹⁴⁰ <https://github.com/sphinx-doc/sphinx/issues/96>

- There is now a `Sphinx.add_lexer()` method to be able to use custom Pygments lexers easily.
- There is now `Sphinx.add_generic_role()` to mirror the Docutils' own function.
- Other changes:
 - Config overrides for single dict keys can now be given on the command line.
 - There is now a `doctest_global_setup` config value that can be used to give setup code for all doctests in the documentation.
 - Source links in HTML are now generated with `rel="nofollow"`.
 - Quickstart can now generate a Windows `make.bat` file.
 - #62⁴¹⁴¹: There is now a `-w` option for `sphinx-build` that writes warnings to a file, in addition to `stderr`.
 - There is now a `-W` option for `sphinx-build` that turns warnings into errors.

14.172 Release 0.5.2 (Mar 24, 2009)

- Properly escape `|` in LaTeX output.
- #71⁴¹⁴²: If a decoding error occurs in source files, print a warning and replace the characters by `"?"`.
- Fix a problem in the HTML search if the index takes too long to load.
- Don't output system messages while resolving, because they would stay in the doctrees even if `keep_warnings` is false.
- #82⁴¹⁴³: Determine the correct path for dependencies noted by docutils. This fixes behavior where a source with dependent files was always reported as changed.
- Recognize toctree directives that are not on section toplevel, but within block items, such as tables.
- Use a new RFC base URL, since `rfc.org` seems down.
- Fix a crash in the todoclist directive when no todo items are defined.
- Don't call LaTeX or dvipng over and over again if it was not found once, and use text-only latex as a substitute in that case.
- Fix problems with footnotes in the LaTeX output.
- Prevent double hyphens becoming en-dashes in literal code in the LaTeX output.
- Open literalinclude files in universal newline mode to allow arbitrary newline conventions.
- Actually make the `-Q` option work.

⁴¹⁴¹ <https://github.com/sphinx-doc/sphinx/issues/62>

⁴¹⁴² <https://github.com/sphinx-doc/sphinx/issues/71>

⁴¹⁴³ <https://github.com/sphinx-doc/sphinx/issues/82>

- [#86⁴¹⁴⁴](#): Fix explicit document titles in toctrees.
- [#81⁴¹⁴⁵](#): Write environment and search index in a manner that is safe from exceptions that occur during dumping.
- [#80⁴¹⁴⁶](#): Fix UnicodeErrors when a locale is set with `setlocale()`.

14.173 Release 0.5.1 (Dec 15, 2008)

- [#67⁴¹⁴⁷](#): Output warnings about failed doctests in the doctest extension even when running in quiet mode.
- [#72⁴¹⁴⁸](#): In `pngmath`, make it possible to give a full path to LaTeX and `dvipng` on Windows. For that to work, the `pngmath_latex` and `pngmath_dvipng` options are no longer split into command and additional arguments; use `pngmath_latex_args` and `pngmath_dvipng_args` to give additional arguments.
- Don't crash on failing doctests with non-ASCII characters.
- Don't crash on writing status messages and warnings containing unencodable characters.
- Warn if a doctest extension block doesn't contain any code.
- Fix the handling of `:param:` and `:type:` doc fields when they contain markup (especially cross-referencing roles).
- [#65⁴¹⁴⁹](#): Fix storage of depth information for PNGs generated by the `pngmath` extension.
- Fix autodoc crash when `automethod` is used outside a class context.
- [#68⁴¹⁵⁰](#): Fix LaTeX writer output for images with specified height.
- [#60⁴¹⁵¹](#): Fix wrong generated image path when including images in sources in subdirectories.
- Fix the JavaScript search when `html_copy_source` is off.
- Fix an indentation problem in autodoc when documenting classes with the option `autoclass_content = "both"` set.
- Don't crash on empty index entries, only emit a warning.
- Fix a typo in the search JavaScript code, leading to unusable search function in some setups.

⁴¹⁴⁴ <https://github.com/sphinx-doc/sphinx/issues/86>

⁴¹⁴⁵ <https://github.com/sphinx-doc/sphinx/issues/81>

⁴¹⁴⁶ <https://github.com/sphinx-doc/sphinx/issues/80>

⁴¹⁴⁷ <https://github.com/sphinx-doc/sphinx/issues/67>

⁴¹⁴⁸ <https://github.com/sphinx-doc/sphinx/issues/72>

⁴¹⁴⁹ <https://github.com/sphinx-doc/sphinx/issues/65>

⁴¹⁵⁰ <https://github.com/sphinx-doc/sphinx/issues/68>

⁴¹⁵¹ <https://github.com/sphinx-doc/sphinx/issues/60>

14.174 Release 0.5 (Nov 23, 2008) -- Birthday release!

New features added

- Markup features:
 - Citations are now global: all citation defined in any file can be referenced from any file. Citations are collected in a bibliography for LaTeX output.
 - Footnotes are now properly handled in the LaTeX builder: they appear at the location of the footnote reference in text, not at the end of a section. Thanks to Andrew McNamara for the initial patch.
 - "System Message" warnings are now automatically removed from the built documentation, and only written to stderr. If you want the old behavior, set the new config value `keep_warnings` to `True`.
 - Glossary entries are now automatically added to the index.
 - Figures with captions can now be referred to like section titles, using the `:ref:` role without an explicit link text.
 - Added `cmember` role for consistency.
 - Lists enumerated by letters or roman numerals are now handled like in standard reST.
 - The `seealso` directive can now also be given arguments, as a short form.
 - You can now document several programs and their options with the new `program` directive.
- HTML output and templates:
 - Incompatible change: The "root" relation link (top left in the relbar) now points to the `master_doc` by default, no longer to a document called "index". The old behavior, while useful in some situations, was somewhat unexpected. Override the "rootrellink" block in the template to customize where it refers to.
 - The JavaScript search now searches for objects before searching in the full text.
 - TOC tree entries now have CSS classes that make it possible to style them depending on their depth.
 - Highlighted code blocks now have CSS classes that make it possible to style them depending on their language.
 - HTML `<meta>` tags via the Docutils `meta`⁴¹⁵² directive are now supported.
 - `SerializingHTMLBuilder` was added as new abstract builder that can be subclassed to serialize build HTML in a specific format. The `PickleHTMLBuilder` is a concrete subclass of it that uses pickle as serialization implementation.
 - `JSONHTMLBuilder` was added as another `SerializingHTMLBuilder` subclass that dumps the generated HTML into JSON files for further processing.

⁴¹⁵² <https://docutils.sourceforge.io/docs/ref/rst/directives.html#meta>

- The `rellinks` block in the layout template is now called `linktags` to avoid confusion with the `relbar` links.
- The HTML builders have two additional attributes now that can be used to disable the anchor-link creation after headlines and definition links.
- Only generate a module index if there are some modules in the documentation.
- New and changed config values:
 - Added support for internationalization in generated text with the `language` and `locale_dirs` config values. Many thanks to language contributors:
 - * Horst Gutmann -- German
 - * Pavel Kosina -- Czech
 - * David Larlet -- French
 - * Michał Kandulski -- Polish
 - * Yasushi Masuda -- Japanese
 - * Guillem Borrell -- Spanish
 - * Luc Saffre and Peter Bertels -- Dutch
 - * Fred Lin -- Traditional Chinese
 - * Roger Demetrescu -- Brazilian Portuguese
 - * Rok Garbas -- Slovenian
 - The new config value `highlight_language` set a global default for highlighting. When `'python3'` is selected, console output blocks are recognized like for `'python'`.
 - Exposed Pygments' lexer guessing as a highlight `"language"` guess.
 - The new config value `latex_elements` allows to override all LaTeX snippets that Sphinx puts into the generated `.tex` file by default.
 - Added `exclude_dirnames` config value that can be used to exclude e.g. CVS directories from source file search.
 - Added `source_encoding` config value to select input encoding.
- Extensions:
 - The new extensions `sphinx.ext.jsmath` and `sphinx.ext.pngmath` provide math support for both HTML and LaTeX builders.
 - The new extension `sphinx.ext.intersphinx` half-automatically creates links to Sphinx documentation of Python objects in other projects.
 - The new extension `sphinx.ext.todo` allows the insertion of "To do" directives whose visibility in the output can be toggled. It also adds a directive to compile a list of all todo items.

- sphinx.ext.autodoc has a new event `autodoc-process-signature` that allows tuning function signature introspection.
- sphinx.ext.autodoc has a new event `autodoc-skip-member` that allows tuning which members are included in the generated content.
- Respect `__all__` when aut documenting module members.
- The `automodule` directive now supports the `synopsis`, `deprecated` and `platform` options.
- Extension API:
 - `Sphinx.add_node()` now takes optional visitor methods for the HTML, LaTeX and text translators; this prevents having to manually patch the classes.
 - Added `Sphinx.add_javascript()` that adds scripts to load in the default HTML template.
 - Added new events: `source-read`, `env-updated`, `env-purge-doc`, `missing-reference`, `build-finished`.
- Other changes:
 - Added a command-line switch `-Q`: it will suppress warnings.
 - Added a command-line switch `-A`: it can be used to supply additional values into the HTML templates.
 - Added a command-line switch `-C`: if it is given, no configuration file `conf.py` is required.
 - Added a distutils command `build_sphinx`: When Sphinx is installed, you can call `python setup.py build_sphinx` for projects that have Sphinx documentation, which will build the docs and place them in the standard distutils build directory.
 - In quickstart, if the selected root path already contains a Sphinx project, complain and abort.

Bugs fixed

- #51⁴¹⁵³: Escape configuration values placed in HTML templates.
- #44⁴¹⁵⁴: Fix small problems in HTML help index generation.
- Fix LaTeX output for line blocks in tables.
- #38⁴¹⁵⁵: Fix "illegal unit" error when using pixel image widths/heights.
- Support table captions in LaTeX output.

⁴¹⁵³ <https://github.com/sphinx-doc/sphinx/issues/51>

⁴¹⁵⁴ <https://github.com/sphinx-doc/sphinx/issues/44>

⁴¹⁵⁵ <https://github.com/sphinx-doc/sphinx/issues/38>

- [#39](#)⁴¹⁵⁶: Work around a bug in Jinja that caused "<generator ...>" to be emitted in HTML output.
- Fix a problem with module links not being generated in LaTeX output.
- Fix the handling of images in different directories.
- [#29](#)⁴¹⁵⁷: Support option lists in the text writer. Make sure that dashes introducing long option names are not contracted to en-dashes.
- Support the "scale" option for images in HTML output.
- [#25](#)⁴¹⁵⁸: Properly escape quotes in HTML help attribute values.
- Fix LaTeX build for some description environments with `:noindex:`.
- [#24](#)⁴¹⁵⁹: Don't crash on uncommon casing of role names (like `:Class:`).
- Only output ANSI colors on color terminals.
- Update to newest fncychap.sty, to fix problems with non-ASCII characters at the start of chapter titles.
- Fix a problem with index generation in LaTeX output, caused by hyperref not being included last.
- Don't disregard return annotations for functions without any parameters.
- Don't throw away labels for code blocks.

14.175 Release 0.4.3 (Oct 8, 2008)

- Fix a bug in autodoc with directly given autodoc members.
- Fix a bug in autodoc that would import a module twice, once as "module", once as "module."
- Fix a bug in the HTML writer that created duplicate id attributes for section titles with Docutils 0.5.
- Properly call `super()` in overridden blocks in templates.
- Add a fix when using XeTeX.
- Unify handling of LaTeX escaping.
- Rebuild everything when the `extensions` config value changes.
- Don't try to remove a nonexistent static directory.
- Fix an indentation problem in production lists.

⁴¹⁵⁶ <https://github.com/sphinx-doc/sphinx/issues/39>

⁴¹⁵⁷ <https://github.com/sphinx-doc/sphinx/issues/29>

⁴¹⁵⁸ <https://github.com/sphinx-doc/sphinx/issues/25>

⁴¹⁵⁹ <https://github.com/sphinx-doc/sphinx/issues/24>

- Fix encoding handling for literal include files: `literalinclude` now has an `encoding` option that defaults to UTF-8.
- Fix the handling of non-ASCII characters entered in quickstart.
- Fix a crash with nonexistent image URIs.

14.176 Release 0.4.2 (Jul 29, 2008)

- Fix rendering of the `samp` role in HTML.
- Fix a bug with LaTeX links to headings leading to a wrong page.
- Reread documents with globbed toctrees when source files are added or removed.
- Add a missing parameter to `PickleHTMLBuilder.handle_page()`.
- Put inheritance info always on its own line.
- Don't automatically enclose code with whitespace in it in quotes; only do this for the `samp` role.
- autodoc now emits a more precise error message when a module can't be imported or an attribute can't be found.
- The JavaScript search now uses the correct file name suffix when referring to found items.
- The `automodule` directive now accepts the `inherited-members` and `show-inheritance` options again.
- You can now rebuild the docs normally after relocating the source and/or doctree directory.

14.177 Release 0.4.1 (Jul 5, 2008)

- Added sub-/superscript node handling to `TextBuilder`.
- Label names in references are now case-insensitive, since reST label names are always lowercased.
- Fix linkcheck builder crash for malformed URLs.
- Add compatibility for admonitions and Docutils 0.5.
- Remove the silly restriction on "rubric" in the LaTeX writer: you can now write arbitrary "rubric" directives, and only those with a title of "Footnotes" will be ignored.
- Copy the HTML logo to the output `_static` directory.
- Fix LaTeX code for modules with underscores in names and platforms.
- Fix a crash with nonlocal image URIs.
- Allow the usage of `:noindex:` in `automodule` directives, as documented.

- Fix the `delete()` docstring processor function in autodoc.
- Fix warning message for nonexistent images.
- Fix JavaScript search in Internet Explorer.

14.178 Release 0.4 (Jun 23, 2008)

New features added

- `tocdepth` can be given as a file-wide metadata entry, and specifies the maximum depth of a TOC of this file.
- The new config value `default_role` can be used to select the default role for all documents.
- Sphinx now interprets field lists with fields like `:param foo:` in description units.
- The new `staticmethod` directive can be used to mark methods as static methods.
- HTML output:
 - The "previous" and "next" links have a more logical structure, so that by following "next" links you can traverse the entire TOC tree.
 - The new event `html-page-context` can be used to include custom values into the context used when rendering an HTML template.
 - Document metadata is now in the default template context, under the name `metadata`.
 - The new config value `html_favicon` can be used to set a favicon for the HTML output. Thanks to Sebastian Wiesner.
 - The new config value `html_use_index` can be used to switch index generation in HTML documents off.
 - The new config value `html_split_index` can be used to create separate index pages for each letter, to be used when the complete index is too large for one page.
 - The new config value `html_short_title` can be used to set a shorter title for the documentation which is then used in the navigation bar.
 - The new config value `html_show_sphinx` can be used to control whether a link to Sphinx is added to the HTML footer.
 - The new config value `html_file_suffix` can be used to set the HTML file suffix to e.g. `.xhtml`.
 - The directories in the `html_static_path` can now contain subdirectories.
 - The module index now isn't collapsed if the number of submodules is larger than the number of toplevel modules.

- The image directive now supports specifying the extension as `.*`, which makes the builder select the one that matches best. Thanks to Sebastian Wiesner.
- The new config value `exclude_trees` can be used to exclude whole subtrees from the search for source files.
- Defaults for configuration values can now be callables, which allows dynamic defaults.
- The new `TextBuilder` creates plain-text output.
- Python 3-style signatures, giving a return annotation via `->`, are now supported.
- Extensions:
 - The autodoc extension now offers a much more flexible way to manipulate docstrings before including them into the output, via the new `autodoc-process-docstring` event.
 - The autodoc extension accepts signatures for functions, methods and classes now that override the signature got via introspection from Python code.
 - The autodoc extension now offers a `show-inheritance` option for autoclass that inserts a list of bases after the signature.
 - The autodoc directives now support the `noindex` flag option.

Bugs fixed

- Correctly report the source location for docstrings included with autodoc.
- Fix the LaTeX output of description units with multiple signatures.
- Handle the figure directive in LaTeX output.
- Handle raw admonitions in LaTeX output.
- Fix determination of the title in HTML help output.
- Handle project names containing spaces.
- Don't write SSI-like comments in HTML output.
- Rename the "sidebar" class to "sphinxsidebar" in order to stay different from reST sidebars.
- Use a binary TOC in HTML help generation to fix issues links without explicit anchors.
- Fix behavior of references to functions/methods with an explicit title.
- Support citation, subscript and superscript nodes in LaTeX writer.
- Provide the standard "class" directive as "cssclass"; else it is shadowed by the Sphinx-defined directive.
- Fix the handling of explicit module names given to autoclass directives. They now show up with the correct module name in the generated docs.
- Enable autodoc to process Unicode docstrings.

- The LaTeX writer now translates line blocks with `\raggedright`, which plays nicer with tables.
- Fix bug with directories in the HTML builder static path.

14.179 Release 0.3 (May 6, 2008)

New features added

- The `toctree` directive now supports a `glob` option that allows glob-style entries in the content.
- If the `pygments_style` config value contains a dot it's treated as the import path of a custom Pygments style class.
- A new config value, `exclude_dirs`, can be used to exclude whole directories from the search for source files.
- The configuration directory (containing `conf.py`) can now be set independently from the source directory. For that, a new command-line option `-c` has been added.
- A new directive `tabularcolumns` can be used to give a tabular column specification for LaTeX output. Tables now use the `tabulary` package. Literal blocks can now be placed in tables, with several caveats.
- A new config value, `latex_use_parts`, can be used to enable parts in LaTeX documents.
- Autodoc now skips inherited members for classes, unless you give the new `inherited-members` option.
- A new config value, `autoclass_content`, selects if the docstring of the class' `__init__` method is added to the directive's body.
- Support for C++ class names (in the style `Class::Function`) in C function descriptions.
- Support for a `toctree_only` item in items for the `latex_documents` config value. This only includes the documents referenced by TOC trees in the output, not the rest of the file containing the directive.

Bugs fixed

- `sphinx.htmlwriter`: Correctly write the TOC file for any structure of the master document. Also encode non-ASCII characters as entities in TOC and index file. Remove two remaining instances of hard-coded "documentation".
- `sphinx.ext.autodoc`: descriptors are detected properly now.
- `sphinx.latexwriter`: implement all reST admonitions, not just `note` and `warning`.
- Lots of little fixes to the LaTeX output and style.

- Fix OpenSearch template and make template URL absolute. The `html_use_opensearch` config value now must give the base URL.
- Some unused files are now stripped from the HTML help file build.

14.180 Release 0.2 (Apr 27, 2008)

Modifiche incompatibili

- Jinja, the template engine used for the default HTML templates, is now no longer shipped with Sphinx. If it is not installed automatically for you (it is now listed as a dependency in `setup.py`), install it manually from PyPI. This will also be needed if you're using Sphinx from a SVN checkout; in that case please also remove the `sphinx/jinja` directory that may be left over from old revisions.
- The clumsy handling of the `index.html` template was removed. The config value `html_index` is gone, and `html_additional_pages` should be used instead. If you need it, the old `index.html` template is still there, called `defindex.html`, and you can port your `html_index` template, using Jinja inheritance, by changing your template:

```
{% extends "defindex.html" %}
{% block tables %}
... old html_index template content ...
{% endblock %}
```

and putting `'index': name of your template in html_additional_pages`.

- In the layout template, redundant blocks were removed; you should use Jinja's standard `{{ super() }}` mechanism instead, as explained in the (newly written) templating docs.

New features added

- Extension API (Application object):
 - Support a new method, `add_crossref_type`. It works like `add_description_unit` but the directive will only create a target and no output.
 - Support a new method, `add_transform`. It takes a standard Docutils Transform subclass which is then applied by Sphinx's reader on parsing reST document trees.
 - Add support for other template engines than Jinja, by adding an abstraction called a "template bridge". This class handles rendering of templates and can be changed using the new configuration value `"template_bridge"`.
 - The config file itself can be an extension (if it provides a `setup()` function).
- Markup:

- New directive, `currentmodule`. It can be used to indicate the module name of the following documented things without creating index entries.
- Allow giving a different title to documents in the toctree.
- Allow giving multiple options in a `cmdoption` directive.
- Fix display of class members without explicit class name given.
- Templates (HTML output):
 - `index.html` renamed to `defindex.html`, see above.
 - There's a new config value, `html_title`, that controls the overall "title" of the set of Sphinx docs. It is used instead everywhere instead of "Projectname vX.Y documentation" now.
 - All references to "documentation" in the templates have been removed, so that it is now easier to use Sphinx for non-documentation documents with the default templates.
 - Templates now have an XHTML doctype, to be consistent with Docutils' HTML output.
 - You can now create an OpenSearch description file with the `html_use_opensearch` config value.
 - You can now quickly include a logo in the sidebar, using the `html_logo` config value.
 - There are new blocks in the sidebar, so that you can easily insert content into the sidebar.
- LaTeX output:
 - The `sphinx.sty` package was cleaned of unused stuff.
 - You can include a logo in the title page with the `latex_logo` config value.
 - You can define the link colors and a border and background color for verbatim environments.

Thanks to Jacob Kaplan-Moss, Talin, Jeroen Ruigrok van der Werven and Sebastian Wiesner for suggestions.

Bugs fixed

- `sphinx.ext.autodoc`: Don't check `__module__` for explicitly given members. Remove "self" in class constructor argument list.
- `sphinx.htmlwriter`: Don't use `os.path` for joining image HREFs.
- `sphinx.htmlwriter`: Don't use `SmartyPants` for HTML attribute values.
- `sphinx.latexwriter`: Implement option lists. Also, some other changes were made to `sphinx.sty` in order to enhance compatibility and remove old unused stuff. Thanks to Gael Varoquaux for that!

- sphinx.roles: Fix referencing glossary terms with explicit targets.
- sphinx.environment: Don't swallow TOC entries when resolving subtrees.
- sphinx.quickstart: Create a sensible default latex_documents setting.
- sphinx.builder, sphinx.environment: Gracefully handle some user error cases.
- sphinx.util: Follow symbolic links when searching for documents.

14.181 Release 0.1.61950 (Mar 26, 2008)

- sphinx.quickstart: Fix format string for Makefile.

14.182 Release 0.1.61945 (Mar 26, 2008)

- sphinx.htmlwriter, sphinx.latexwriter: Support the `.. image::` directive by copying image files to the output directory.
- sphinx.builder: Consistently name "special" HTML output directories with a leading underscore; this means `_sources` and `_static`.
- sphinx.environment: Take dependent files into account when collecting the set of outdated sources.
- sphinx.directives: Record files included with `.. literalinclude::` as dependencies.
- sphinx.ext.autodoc: Record files from which docstrings are included as dependencies.
- sphinx.builder: Rebuild all HTML files in case of a template change.
- sphinx.builder: Handle unavailability of TOC relations (previous/ next chapter) more gracefully in the HTML builder.
- sphinx.latexwriter: Include `fncychap.sty` which doesn't seem to be very common in TeX distributions. Add a `clean` target in the latex Makefile. Really pass the correct paper and size options to the LaTeX document class.
- setup: On Python 2.4, don't egg-depend on Docutils if a Docutils is already installed -- else it will be overwritten.

14.183 Release 0.1.61843 (Mar 24, 2008)

- sphinx.quickstart: Really don't create a makefile if the user doesn't want one.
- setup: Don't install scripts twice, via setuptools entry points and distutils scripts. Only install via entry points.
- sphinx.builder: Don't recognize the HTML builder's copied source files (under `_sources`) as input files if the source suffix is `.txt`.

- sphinx.highlighting: Generate correct markup for LaTeX Verbatim environment escapes even if Pygments is not installed.
- sphinx.builder: The WebHTMLBuilder is now called PickleHTMLBuilder.
- sphinx.htmlwriter: Make parsed-literal blocks work as expected, not highlighting them via Pygments.
- sphinx.environment: Don't error out on reading an empty source file.

14.184 Release 0.1.61798 (Mar 23, 2008)

- sphinx: Work with Docutils SVN snapshots as well as 0.4.
- sphinx.ext.doctest: Make the group in which doctest blocks are placed selectable, and default to 'default'.
- sphinx.ext.doctest: Replace <BLANKLINE> in doctest blocks by real blank lines for presentation output, and remove doctest options given inline.
- sphinx.environment: Move doctest_blocks out of block_quotes to support indented doctest blocks.
- sphinx.ext.autodoc: Render .. automodule:: docstrings in a section node, so that module docstrings can contain proper sectioning.
- sphinx.ext.autodoc: Use the module's encoding for decoding docstrings, rather than requiring ASCII.

14.185 Release 0.1.61611 (Mar 21, 2008)

- First public release.

Progetti che utilizzano Sphinx

Questo è un elenco alfabetico (incompleto) di progetti che utilizzano Sphinx o ne stanno sperimentando l'utilizzo per la loro documentazione. Per essere inclusi, inviare una mail a [the Google group](#)⁴¹⁶⁰.

L'elenco è raggruppato in sezioni per facilitare la ricerca di esempi interessanti.

15.1 Documentazione che utilizza il tema alabaster

- [Alabaster](#)⁴¹⁶¹
- [Blinker](#)⁴¹⁶²
- [Calibre](#)⁴¹⁶³
- [CherryPy](#)⁴¹⁶⁴
- [Click](#)⁴¹⁶⁵ (customized)
- [coala](#)⁴¹⁶⁶ (customized)
- [CodePy](#)⁴¹⁶⁷
- [Django Q](#)⁴¹⁶⁸

⁴¹⁶⁰ <https://groups.google.com/forum/#!forum/sphinx-users>

⁴¹⁶¹ <https://alabaster.readthedocs.io/>

⁴¹⁶² <https://blinker.readthedocs.io/>

⁴¹⁶³ <https://manual.calibre-ebook.com/>

⁴¹⁶⁴ <https://cherrypy.readthedocs.io/>

⁴¹⁶⁵ <https://click.palletsprojects.com/>

⁴¹⁶⁶ <https://docs.coala.io/>

⁴¹⁶⁷ <https://documen.tician.de/codepy/>

⁴¹⁶⁸ <https://django-q.readthedocs.io/>

- Eve⁴¹⁶⁹ (Python REST API framework)
- Fabric⁴¹⁷⁰
- Fityk⁴¹⁷¹
- Flask⁴¹⁷²
- Flask-OpenID⁴¹⁷³
- Invoke⁴¹⁷⁴
- Jinja⁴¹⁷⁵
- Lino⁴¹⁷⁶ (customized)
- Linux kernel⁴¹⁷⁷ (customized)
- marbl⁴¹⁷⁸
- MeshPy⁴¹⁷⁹
- Molecule⁴¹⁸⁰
- Momotor LTI⁴¹⁸¹
- Podman⁴¹⁸²
- PyCUDA⁴¹⁸³
- PyOpenCL⁴¹⁸⁴
- PyLangAcq⁴¹⁸⁵
- pytest⁴¹⁸⁶ (customized)
- python-apt⁴¹⁸⁷
- PyVisfile⁴¹⁸⁸

⁴¹⁶⁹ <https://docs.python-eve.org/>

⁴¹⁷⁰ <https://docs.fabfile.org/>

⁴¹⁷¹ <https://fityk.nieto.pl/>

⁴¹⁷² <https://flask.palletsprojects.com/>

⁴¹⁷³ <https://pythonhosted.org/Flask-OpenID/>

⁴¹⁷⁴ <https://docs.pyinvoke.org/>

⁴¹⁷⁵ <https://jinja.palletsprojects.com/>

⁴¹⁷⁶ <https://www.lino-framework.org/>

⁴¹⁷⁷ <https://www.kernel.org/doc/html/latest/index.html>

⁴¹⁷⁸ <https://getmarbl.readthedocs.io/>

⁴¹⁷⁹ <https://documen.tician.de/meshpy/>

⁴¹⁸⁰ <https://molecule.readthedocs.io/>

⁴¹⁸¹ <https://momotor.org/doc/lti/canvas/>

⁴¹⁸² <https://docs.podman.io/>

⁴¹⁸³ <https://documen.tician.de/pycuda/>

⁴¹⁸⁴ <https://documen.tician.de/pyopencl/>

⁴¹⁸⁵ <https://pylangacq.org/>

⁴¹⁸⁶ <https://docs.pytest.org/>

⁴¹⁸⁷ <https://apt-team.pages.debian.net/python-apt/>

⁴¹⁸⁸ <https://documen.tician.de/pyvisfile/>

- [Requests](#)⁴¹⁸⁹
- [searx](#)⁴¹⁹⁰
- [Spyder](#)⁴¹⁹¹ (customized)
- [Tablib](#)⁴¹⁹²
- [urllib3](#)⁴¹⁹³ (customized)
- [Werkzeug](#)⁴¹⁹⁴
- [Write the Docs](#)⁴¹⁹⁵

15.2 Documentazione che utilizza il tema classic

- [Advanced Generic Widgets](#)⁴¹⁹⁶ (customized)
- [Apache CouchDB](#)⁴¹⁹⁷ (customized)
- [APSW](#)⁴¹⁹⁸
- [Arb](#)⁴¹⁹⁹
- [Beautiful Soup](#)⁴²⁰⁰
- [Blender API](#)⁴²⁰¹
- [Bugzilla](#)⁴²⁰²
- [Buildbot](#)⁴²⁰³
- [CMake](#)⁴²⁰⁴ (customized)
- [Chaco](#)⁴²⁰⁵ (customized)
- [DEAP](#)⁴²⁰⁶ (customized)

⁴¹⁸⁹ <https://requests.readthedocs.io/>

⁴¹⁹⁰ <https://asciimoo.github.io/searx/>

⁴¹⁹¹ <https://docs.spyder-ide.org/>

⁴¹⁹² <http://docs.python-tablib.org/>

⁴¹⁹³ <https://urllib3.readthedocs.io/>

⁴¹⁹⁴ <https://werkzeug.palletsprojects.com/>

⁴¹⁹⁵ <https://www.writethedocs.org/>

⁴¹⁹⁶ https://xoomer.virgilio.it/infinity77/AGW_Docs/

⁴¹⁹⁷ <https://docs.couchdb.org/>

⁴¹⁹⁸ <https://rogerbinns.github.io/apsw/>

⁴¹⁹⁹ <https://arblib.org/>

⁴²⁰⁰ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

⁴²⁰¹ <https://docs.blender.org/api/current/>

⁴²⁰² <https://bugzilla.readthedocs.io/>

⁴²⁰³ <https://docs.buildbot.net/latest/>

⁴²⁰⁴ <https://cmake.org/documentation/>

⁴²⁰⁵ <https://docs.enthought.com/chaco/>

⁴²⁰⁶ <https://deap.readthedocs.io/>

- [Director](#)⁴²⁰⁷
- [EZ-Draw](#)⁴²⁰⁸ (personalizzato)
- [Generic Mapping Tools \(GMT\)](#)⁴²⁰⁹ (customized)
- [Genomedata](#)⁴²¹⁰
- [GetFEM](#)⁴²¹¹ (personalizzato)
- [Glasgow Haskell Compiler](#)⁴²¹² (personalizzato)
- [Grok](#)⁴²¹³ (personalizzato)
- [GROMACS](#)⁴²¹⁴
- [GSL Shell](#)⁴²¹⁵
- [Hands-on Python Tutorial](#)⁴²¹⁶
- [Kaa](#)⁴²¹⁷ (customized)
- [Leo](#)⁴²¹⁸ (customized)
- [Mayavi](#)⁴²¹⁹ (customized)
- [MediaGoblin](#)⁴²²⁰ (customized)
- [mpmath](#)⁴²²¹
- [OpenCV](#)⁴²²² (customized)
- [OpenEXR](#)⁴²²³
- [OpenGDA](#)⁴²²⁴
- [phpDocumentor](#)⁴²²⁵ (customized)
- [Plone](#)⁴²²⁶ (customized)

⁴²⁰⁷ <https://pythonhosted.org/director/>

⁴²⁰⁸ <https://pageperso.lis-lab.fr/~edouard.thiel/ez-draw/doc/en/html/ez-manual.html>

⁴²⁰⁹ <https://docs.generic-mapping-tools.org/latest/>

⁴²¹⁰ <https://noble.gs.washington.edu/proj/genomedata/doc/1.3.3/>

⁴²¹¹ <https://getfem.org/>

⁴²¹² https://downloads.haskell.org/~ghc/latest/docs/html/users_guide/

⁴²¹³ <https://web.archive.org/web/20230708190705/http://grok.zope.org/doc/current/>

⁴²¹⁴ <https://manual.gromacs.org/documentation/>

⁴²¹⁵ <https://www.nongnu.org/gsl-shell/>

⁴²¹⁶ <http://anh.cs.luc.edu:80/python/hands-on/3.1/handsonHtml/>

⁴²¹⁷ <https://freevo.github.io/kaa-base/>

⁴²¹⁸ <https://leo-editor.github.io/leo-editor/>

⁴²¹⁹ <https://docs.enthought.com/mayavi/mayavi/>

⁴²²⁰ <https://mediagoblin.readthedocs.io/>

⁴²²¹ <https://mpmath.org/doc/current/>

⁴²²² <https://docs.opencv.org/>

⁴²²³ <https://excamera.com/articles/26/doc/index.html>

⁴²²⁴ <https://alfred.diamond.ac.uk/documentation/>

⁴²²⁵ <https://docs.phpdoc.org/>

⁴²²⁶ <https://docs.plone.org/>

- [PyEMD](#)⁴²²⁷
- [Pyevolve](#)⁴²²⁸
- [Pygame](#)⁴²²⁹ (customized)
- [PyMQI](#)⁴²³⁰
- [PyQt4](#)⁴²³¹ (personalizzato)
- [PyQt5](#)⁴²³² (personalizzato)
- [Python 2](#)⁴²³³
- [Python 3](#)⁴²³⁴ (customized)
- [Python Packaging Authority](#)⁴²³⁵ (customized)
- [Ring programming language](#)⁴²³⁶ (customized)
- [SageMath](#)⁴²³⁷ (customized)
- [Segway](#)⁴²³⁸
- [simuPOP](#)⁴²³⁹ (customized)
- [SymPy](#)⁴²⁴⁰
- [TurboGears](#)⁴²⁴¹ (customized)
- [vtk](#)⁴²⁴²
- [Varnish](#)⁴²⁴³ (customized, alabaster for index)
- [Waf](#)⁴²⁴⁴
- [wxPython Phoenix](#)⁴²⁴⁵ (customized)
- [z3c](#)⁴²⁴⁶

⁴²²⁷ <https://pyemd.readthedocs.io/>

⁴²²⁸ <https://pyevolve.sourceforge.net/>

⁴²²⁹ <https://www.pygame.org/docs/>

⁴²³⁰ <https://dsuch.github.io/pymqi/>

⁴²³¹ <https://pyqt.sourceforge.net/Docs/PyQt4/>

⁴²³² <https://pyqt.sourceforge.net/Docs/PyQt5/>

⁴²³³ <https://docs.python.org/2/>

⁴²³⁴ <https://docs.python.org/3/>

⁴²³⁵ <https://www.pypa.io/>

⁴²³⁶ <https://ring-lang.github.io/doc1.20/>

⁴²³⁷ <https://doc.sagemath.org/>

⁴²³⁸ <https://noble.gs.washington.edu/proj/segway/doc/1.1.0/segway.html>

⁴²³⁹ <https://bopeng.github.io/simuPOP/>

⁴²⁴⁰ <https://docs.sympy.org/>

⁴²⁴¹ <https://turbogears.readthedocs.io/>

⁴²⁴² <https://docs.entthought.com/mayavi/tvtk/>

⁴²⁴³ <https://www.varnish-cache.org/docs/>

⁴²⁴⁴ <https://waf.io/apidocs/>

⁴²⁴⁵ <https://wxpython.org/Phoenix/docs/html/main.html>

⁴²⁴⁶ <https://www.ibiblio.org/paulcarduner/z3ctutorial/>

- [zc.async](#)⁴²⁴⁷ (customized)
- [Zope](#)⁴²⁴⁸ (personalizzato)

15.3 Documentazione che utilizza il tema sphinxdoc

- [ABRT](#)⁴²⁴⁹
- [cartopy](#)⁴²⁵⁰
- [Jython](#)⁴²⁵¹
- [LLVM](#)⁴²⁵²
- [PyCantonese](#)⁴²⁵³
- [Pyre](#)⁴²⁵⁴
- [pySPACE](#)⁴²⁵⁵
- [Pysparse](#)⁴²⁵⁶
- [PyTango](#)⁴²⁵⁷
- [Python Wild Magic](#)⁴²⁵⁸ (customized)
- [RDKit](#)⁴²⁵⁹
- [Reteisi](#)⁴²⁶⁰ (personalizzato)
- [Turbulenz](#)⁴²⁶¹

⁴²⁴⁷ <https://pythonhosted.org/zc.async/>

⁴²⁴⁸ <https://www.zope.dev/>

⁴²⁴⁹ <https://abrt.readthedocs.io/>

⁴²⁵⁰ <https://scitools.org.uk/cartopy/docs/latest/>

⁴²⁵¹ <https://jython.readthedocs.io/>

⁴²⁵² <https://llvm.org/docs/>

⁴²⁵³ <https://pycantonese.org/>

⁴²⁵⁴ <https://pyre.readthedocs.io/>

⁴²⁵⁵ <https://pyspace.github.io/pyspace/>

⁴²⁵⁶ <https://pysparse.sourceforge.net/>

⁴²⁵⁷ <https://pytango.readthedocs.io>

⁴²⁵⁸ <https://vmlaker.github.io/pythonwildmagic/>

⁴²⁵⁹ <https://www.rdkit.org/docs/>

⁴²⁶⁰ <https://www.reteisi.org/contents.html>

⁴²⁶¹ <http://docs.turbulenz.com/>

15.4 Documentazione che utilizza il tema nature

- Alembic⁴²⁶²
- Cython⁴²⁶³
- easybuild⁴²⁶⁴
- libLAS⁴²⁶⁵ (customized)
- Lmod⁴²⁶⁶
- MapServer⁴²⁶⁷ (customized)
- PyWavelets⁴²⁶⁸
- Setuptools⁴²⁶⁹
- Spring Python⁴²⁷⁰
- StatsModels⁴²⁷¹ (customized)
- Sylli⁴²⁷²

15.5 Documentazione che utilizza un altro tema nativo

- Breathe⁴²⁷³ (haiku)
- Breezy (fork of Bazaar)⁴²⁷⁴ (agogo)
- MPipe⁴²⁷⁵ (sphinx13)
- NLTK⁴²⁷⁶ (agogo)
- PyPubSub⁴²⁷⁷ (bizstyle)
- Pylons⁴²⁷⁸ (pyramid)

⁴²⁶² <https://alembic.sqlalchemy.org/>

⁴²⁶³ <https://docs.cython.org/>

⁴²⁶⁴ <https://easybuild.readthedocs.io/>

⁴²⁶⁵ <https://liblas.org/>

⁴²⁶⁶ <https://lmod.readthedocs.io/>

⁴²⁶⁷ <https://mapserver.org/>

⁴²⁶⁸ <https://pywavelets.readthedocs.io/>

⁴²⁶⁹ <https://setuptools.readthedocs.io/>

⁴²⁷⁰ <https://docs.spring.io/spring-python/1.2.x/sphinx/html/>

⁴²⁷¹ <https://www.statsmodels.org/>

⁴²⁷² <https://sylli.sourceforge.net/>

⁴²⁷³ <https://breathe.readthedocs.io/>

⁴²⁷⁴ <https://www.breezy-vcs.org/doc/en/>

⁴²⁷⁵ <https://vmlaker.github.io/mpipe/>

⁴²⁷⁶ <https://www.nltk.org/>

⁴²⁷⁷ <https://pypubsub.readthedocs.io/>

⁴²⁷⁸ <https://docs.pylonsproject.org/projects/pylons-webframework/>

- Pyramid web framework⁴²⁷⁹ (pyramid)
- RxDock⁴²⁸⁰ (bizstyle)
- Sphinx⁴²⁸¹ (sphinx13) :-)
- Valence⁴²⁸² (haiku, customized)

15.6 Documentazione che utilizza sphinx_rtd_theme

- Aesara (fork of Theano)⁴²⁸³
- Annotator⁴²⁸⁴
- Ansible⁴²⁸⁵ (customized)
- Arcade⁴²⁸⁶
- aria2⁴²⁸⁷
- ASE⁴²⁸⁸
- asvin⁴²⁸⁹
- Autofac⁴²⁹⁰
- BigchainDB⁴²⁹¹
- Blender Reference Manual⁴²⁹²
- Blocks⁴²⁹³
- bootstrap-datepicker⁴²⁹⁴
- Certbot⁴²⁹⁵
- CKAN⁴²⁹⁶

⁴²⁷⁹ <https://docs.pylonsproject.org/projects/pyramid/>

⁴²⁸⁰ <https://rxdock.gitlab.io/documentation/devel/html/>

⁴²⁸¹ <https://www.sphinx-doc.org/>

⁴²⁸² <https://docs.valence.desire2learn.com/>

⁴²⁸³ <https://aesara.readthedocs.io/>

⁴²⁸⁴ <https://docs.annotatorjs.org/>

⁴²⁸⁵ <https://docs.ansible.com/>

⁴²⁸⁶ <https://arcade.academy/>

⁴²⁸⁷ <https://aria2.github.io/manual/en/html/>

⁴²⁸⁸ <https://wiki.fysik.dtu.dk/ase/>

⁴²⁸⁹ <https://asvin.readthedocs.io/>

⁴²⁹⁰ <https://docs.autofac.org/>

⁴²⁹¹ <https://docs.bigchaindb.com/>

⁴²⁹² <https://docs.blender.org/manual/>

⁴²⁹³ <https://blocks.readthedocs.io/>

⁴²⁹⁴ <https://bootstrap-datepicker.readthedocs.io/>

⁴²⁹⁵ <https://certbot.eff.org/docs/>

⁴²⁹⁶ <https://docs.ckan.org/>

- Copr Buildsystem⁴²⁹⁷ (customized)
- Coreboot⁴²⁹⁸
- Chainer⁴²⁹⁹ (customized)
- citeproc-js⁴³⁰⁰
- cloud-init⁴³⁰¹
- CodeIgniter⁴³⁰²
- Conda⁴³⁰³
- Corda⁴³⁰⁴
- Dask⁴³⁰⁵
- Databricks⁴³⁰⁶ (customized)
- Dataiku DSS⁴³⁰⁷
- DNF⁴³⁰⁸
- Distro Tracker⁴³⁰⁹
- Django-cas-ng⁴³¹⁰
- dj-stripe⁴³¹¹
- edX⁴³¹²
- Electrum⁴³¹³
- ESWP3⁴³¹⁴
- Ethereum Homestead⁴³¹⁵
- Exhale⁴³¹⁶

⁴²⁹⁷ <https://docs.pagure.org/copr.copr/>

⁴²⁹⁸ <https://doc.coreboot.org/>

⁴²⁹⁹ <https://docs.chainer.org/>

⁴³⁰⁰ <https://citeproc-js.readthedocs.io/>

⁴³⁰¹ <https://cloudinit.readthedocs.io/>

⁴³⁰² https://www.codeigniter.com/user_guide/

⁴³⁰³ <https://conda.io/docs/>

⁴³⁰⁴ <https://docs.corda.net/>

⁴³⁰⁵ <https://dask.pydata.org/>

⁴³⁰⁶ <https://docs.databricks.com/>

⁴³⁰⁷ <https://doc.dataiku.com/>

⁴³⁰⁸ <https://dnf.readthedocs.io/>

⁴³⁰⁹ <https://qa.pages.debian.net/distro-tracker/>

⁴³¹⁰ <https://djangocas.dev/docs/>

⁴³¹¹ <https://dj-stripe.github.io/dj-stripe/>

⁴³¹² <https://docs.edx.org/>

⁴³¹³ <https://docs.electrum.org/>

⁴³¹⁴ <https://eswp3.readthedocs.io/>

⁴³¹⁵ <https://www.ethdocs.org/>

⁴³¹⁶ <https://exhale.readthedocs.io/>

- [Faker](#)⁴³¹⁷
- [Fidimag](#)⁴³¹⁸
- [Flake8](#)⁴³¹⁹
- [Flatpak](#)⁴³²⁰
- [FluidDyn](#)⁴³²¹
- [Fluidsim](#)⁴³²²
- [Gallium](#)⁴³²³
- [GeoNode](#)⁴³²⁴
- [Glances](#)⁴³²⁵
- [Godot](#)⁴³²⁶
- [Graylog](#)⁴³²⁷
- [GPAW](#)⁴³²⁸ (customized)
- [HDF5 for Python \(h5py\)](#)⁴³²⁹
- [HyperKitty](#)⁴³³⁰
- [Hyperledger Fabric](#)⁴³³¹
- [IdentityServer](#)⁴³³²
- [Idris](#)⁴³³³
- [Inkscape](#)⁴³³⁴ (customized)
- [javasphinx](#)⁴³³⁵
- [Jupyter Notebook](#)⁴³³⁶

⁴³¹⁷ <https://faker.readthedocs.io/>

⁴³¹⁸ <https://fidimag.readthedocs.io/>

⁴³¹⁹ <https://flake8.pycqa.org/>

⁴³²⁰ <https://docs.flatpak.org/>

⁴³²¹ <https://fluiddyn.readthedocs.io/>

⁴³²² <https://fluidsim.readthedocs.io/>

⁴³²³ <https://gallium.readthedocs.io/>

⁴³²⁴ <https://docs.geonode.org/>

⁴³²⁵ <https://glances.readthedocs.io/>

⁴³²⁶ <https://godot.readthedocs.io/>

⁴³²⁷ <https://docs.graylog.org/>

⁴³²⁸ <https://wiki.fysik.dtu.dk/gpaw/>

⁴³²⁹ <https://docs.h5py.org/>

⁴³³⁰ <https://hyperkitty.readthedocs.io/>

⁴³³¹ <https://hyperledger-fabric.readthedocs.io/>

⁴³³² <https://docs.identityserver.io/>

⁴³³³ <https://docs.idris-lang.org/>

⁴³³⁴ <https://inkscape-manuals.readthedocs.io/>

⁴³³⁵ <https://bronto-javasphinx.readthedocs.io/>

⁴³³⁶ <https://jupyter-notebook.readthedocs.io/>

- [Kanboard](#)⁴³³⁷
- [Lasagne](#)⁴³³⁸
- [latexindent.pl](#)⁴³³⁹
- [Learning Apache Spark with Python](#)⁴³⁴⁰
- [LibCEED](#)⁴³⁴¹
- [Linguistica](#)⁴³⁴²
- [Mailman](#)⁴³⁴³
- [MathJax](#)⁴³⁴⁴
- [MDTraj](#)⁴³⁴⁵ (personalizzato)
- [Mesa 3D](#)⁴³⁴⁶
- [micca - MICrobial Community Analysis](#)⁴³⁴⁷
- [MicroPython](#)⁴³⁴⁸
- [Mink](#)⁴³⁴⁹
- [Mockery](#)⁴³⁵⁰
- [mod_wsgi](#)⁴³⁵¹
- [MoinMoin](#)⁴³⁵²
- [Mopidy](#)⁴³⁵³
- [mpi4py](#)⁴³⁵⁴
- [MyHDL](#)⁴³⁵⁵
- [Mypy](#)⁴³⁵⁶

⁴³³⁷ <https://docs.kanboard.org/>

⁴³³⁸ <https://lasagne.readthedocs.io/>

⁴³³⁹ <https://latexindentpl.readthedocs.io/>

⁴³⁴⁰ <https://runawayhorse001.github.io/LearningApacheSpark>

⁴³⁴¹ <https://libceed.readthedocs.io/>

⁴³⁴² <https://linguistica-uchicago.github.io/lxa5/>

⁴³⁴³ <https://docs.list.org/>

⁴³⁴⁴ <https://docs.mathjax.org/>

⁴³⁴⁵ <https://mdtraj.org/>

⁴³⁴⁶ <https://docs.mesa3d.org/>

⁴³⁴⁷ <https://micca.readthedocs.io/>

⁴³⁴⁸ <https://docs.micropython.org/>

⁴³⁴⁹ <https://mink.behat.org/>

⁴³⁵⁰ <https://docs.mockery.io/>

⁴³⁵¹ <https://modwsgi.readthedocs.io/>

⁴³⁵² <https://moin-20.readthedocs.io/>

⁴³⁵³ <https://docs.mopidy.com/>

⁴³⁵⁴ <https://mpi4py.readthedocs.io/>

⁴³⁵⁵ <https://docs.myhdl.org/>

⁴³⁵⁶ <https://mypy.readthedocs.io/>

- [Netgate Docs](#)⁴³⁵⁷
- [Nextcloud Server](#)⁴³⁵⁸
- [Nextflow](#)⁴³⁵⁹
- [nghttp2](#)⁴³⁶⁰
- [NICOS](#)⁴³⁶¹ (customized)
- [OpenFAST](#)⁴³⁶²
- [Panda3D](#)⁴³⁶³ (customized)
- [Pelican](#)⁴³⁶⁴
- [picamera](#)⁴³⁶⁵
- [Pillow](#)⁴³⁶⁶
- [pip](#)⁴³⁶⁷
- [Paver](#)⁴³⁶⁸
- [peewee](#)⁴³⁶⁹
- [Phinx](#)⁴³⁷⁰
- [phpMyAdmin](#)⁴³⁷¹
- [PHPUnit](#)⁴³⁷²
- [PHPWord](#)⁴³⁷³
- [PROS](#)⁴³⁷⁴ (customized)
- [Pweave](#)⁴³⁷⁵
- [pyca/cryptography](#)⁴³⁷⁶

⁴³⁵⁷ <https://docs.netgate.com/>

⁴³⁵⁸ <https://docs.nextcloud.com/#server>

⁴³⁵⁹ <https://www.nextflow.io/docs/latest/index.html>

⁴³⁶⁰ <https://nghttp2.org/documentation/>

⁴³⁶¹ <https://forge.frm2.tum.de/nicos/doc/nicos-master/>

⁴³⁶² <https://openfast.readthedocs.io/>

⁴³⁶³ <https://docs.panda3d.org/>

⁴³⁶⁴ <https://docs.getpelican.com/>

⁴³⁶⁵ <https://picamera.readthedocs.io/>

⁴³⁶⁶ <https://pillow.readthedocs.io/>

⁴³⁶⁷ <https://pip.pypa.io/>

⁴³⁶⁸ <https://paver.readthedocs.io/>

⁴³⁶⁹ <https://docs.peewee-orm.com/>

⁴³⁷⁰ <https://docs.phinx.org/>

⁴³⁷¹ <https://docs.phpmyadmin.net/>

⁴³⁷² <https://phpunit.readthedocs.io/>

⁴³⁷³ <https://phpword.readthedocs.io/>

⁴³⁷⁴ <https://pros.cs.purdue.edu/v5/>

⁴³⁷⁵ <https://mpastell.com/pweave/>

⁴³⁷⁶ <https://cryptography.io/>

- [pyglet](#)⁴³⁷⁷
- [PyNaCl](#)⁴³⁷⁸
- [pyOpenSSL](#)⁴³⁷⁹
- [PyPy](#)⁴³⁸⁰
- [python-sqlparse](#)⁴³⁸¹
- [PyVISA](#)⁴³⁸²
- [Read The Docs](#)⁴³⁸³
- [RenderDoc](#)⁴³⁸⁴
- [ROCm Platform](#)⁴³⁸⁵
- [Free your information from their silos \(French\)](#)⁴³⁸⁶ (customized)
- [Releases Sphinx extension](#)⁴³⁸⁷
- [Qtile](#)⁴³⁸⁸
- [Quex](#)⁴³⁸⁹
- [QuTiP](#)⁴³⁹⁰
- [Sawtooth](#)⁴³⁹¹
- [Scapy](#)⁴³⁹²
- [SimGrid](#)⁴³⁹³
- [SimPy](#)⁴³⁹⁴
- [six](#)⁴³⁹⁵
- [Solidity](#)⁴³⁹⁶

⁴³⁷⁷ <https://pyglet.readthedocs.io/>

⁴³⁷⁸ <https://pynacl.readthedocs.io/>

⁴³⁷⁹ <https://www.pyopenssl.org/>

⁴³⁸⁰ <https://doc.pypy.org/>

⁴³⁸¹ <https://sqlparse.readthedocs.io/>

⁴³⁸² <https://pyvisa.readthedocs.io/>

⁴³⁸³ <https://docs.readthedocs.io/>

⁴³⁸⁴ <https://renderdoc.org/docs/>

⁴³⁸⁵ <https://rocmdocs.amd.com/>

⁴³⁸⁶ <https://redaction-technique.org/>

⁴³⁸⁷ <https://releases.readthedocs.io/>

⁴³⁸⁸ <https://docs.qtile.org/>

⁴³⁸⁹ <https://quex.sourceforge.net/doc/html/main.html>

⁴³⁹⁰ <https://qutip.readthedocs.io/en/latest/>

⁴³⁹¹ <https://sawtooth.splinter.dev/docs>

⁴³⁹² <https://scapy.readthedocs.io/>

⁴³⁹³ <https://simgrid.org/doc/latest/>

⁴³⁹⁴ <https://simpy.readthedocs.io/>

⁴³⁹⁵ <https://six.readthedocs.io/>

⁴³⁹⁶ <https://solidity.readthedocs.io/>

- [Sonos Controller \(SoCo\)](#)⁴³⁹⁷
- [Sphinx AutoAPI](#)⁴³⁹⁸
- [sphinx-argparse](#)⁴³⁹⁹
- [sphinx-tabs](#)⁴⁴⁰⁰
- [Sphinx-Gallery](#)⁴⁴⁰¹ (customized)
- [Sphinx with Github Webpages](#)⁴⁴⁰²
- [SpotBugs](#)⁴⁴⁰³
- [StarUML](#)⁴⁴⁰⁴
- [Sublime Text Unofficial Documentation](#)⁴⁴⁰⁵
- [SunPy](#)⁴⁴⁰⁶
- [Sylius](#)⁴⁴⁰⁷
- [Syncthing](#)⁴⁴⁰⁸
- [Tango Controls](#)⁴⁴⁰⁹ (customized)
- [ThreatConnect](#)⁴⁴¹⁰
- [TrueNAS](#)⁴⁴¹¹ (customized)
- [Tuleap](#)⁴⁴¹²
- [TYPO3](#)⁴⁴¹³ (customized)
- [Veyon](#)⁴⁴¹⁴
- [Ubiquity](#)⁴⁴¹⁵
- [uWSGI](#)⁴⁴¹⁶

⁴³⁹⁷ <https://docs.python-soco.com/>

⁴³⁹⁸ <https://sphinx-autoapi.readthedocs.io/>

⁴³⁹⁹ <https://sphinx-argparse.readthedocs.io/>

⁴⁴⁰⁰ <https://sphinx-tabs.readthedocs.io/>

⁴⁴⁰¹ <https://sphinx-gallery.readthedocs.io/>

⁴⁴⁰² <https://runawayhorse001.github.io/SphinxGithub>

⁴⁴⁰³ <https://spotbugs.readthedocs.io/>

⁴⁴⁰⁴ <https://docs.staruml.io/>

⁴⁴⁰⁵ <https://docs.sublimetext.info/>

⁴⁴⁰⁶ <https://docs.sunpy.org/>

⁴⁴⁰⁷ <https://docs.sylius.com/>

⁴⁴⁰⁸ <https://docs.syncthing.net/>

⁴⁴⁰⁹ <https://tango-controls.readthedocs.io/>

⁴⁴¹⁰ <https://docs.threatconnect.com/>

⁴⁴¹¹ <https://www.ixsystems.com/documentation/truenas/>

⁴⁴¹² <https://tuleap.net/doc/en/>

⁴⁴¹³ <https://docs.typo3.org/>

⁴⁴¹⁴ <https://docs.veyon.io/>

⁴⁴¹⁵ <https://micro-framework.readthedocs.io/>

⁴⁴¹⁶ <https://uwsgi-docs.readthedocs.io/>

- [virtualenv](#)⁴⁴¹⁷
- [Wagtail](#)⁴⁴¹⁸
- [Web Application Attack and Audit Framework \(w3af\)](#)⁴⁴¹⁹
- [Weblate](#)⁴⁴²⁰
- [x265](#)⁴⁴²¹
- [Zeek](#)⁴⁴²²
- [Zulip](#)⁴⁴²³

15.7 Documentazione che utilizza sphinx_bootstrap_theme

- [Bootstrap Theme](#)⁴⁴²⁴
- [C/C++ Software Development with Eclipse](#)⁴⁴²⁵
- [Dataverse](#)⁴⁴²⁶
- [e-cidadania](#)⁴⁴²⁷
- [Hangfire](#)⁴⁴²⁸
- [Hedge](#)⁴⁴²⁹
- [ObsPy](#)⁴⁴³⁰
- [OPNFV](#)⁴⁴³¹
- [Pootle](#)⁴⁴³²
- [PyUblas](#)⁴⁴³³
- [seaborn](#)⁴⁴³⁴

⁴⁴¹⁷ <https://virtualenv.readthedocs.io/>

⁴⁴¹⁸ <https://docs.wagtail.io/>

⁴⁴¹⁹ <https://docs.w3af.org/>

⁴⁴²⁰ <https://docs.weblate.org/>

⁴⁴²¹ <https://x265.readthedocs.io/>

⁴⁴²² <https://docs.zeek.org/>

⁴⁴²³ <https://zulip.readthedocs.io/>

⁴⁴²⁴ <https://ryan-roemer.github.io/sphinx-bootstrap-theme/>

⁴⁴²⁵ <https://eclipsebook.in/>

⁴⁴²⁶ <https://guides.dataverse.org/>

⁴⁴²⁷ <https://e-cidadania.readthedocs.io/>

⁴⁴²⁸ <https://docs.hangfire.io/>

⁴⁴²⁹ <https://documen.tician.de/hedge/>

⁴⁴³⁰ <https://docs.obspy.org/>

⁴⁴³¹ <https://docs.opnfv.org/>

⁴⁴³² <https://docs.translatehouse.org/projects/pootle/>

⁴⁴³³ <https://documen.tician.de/pyublas/>

⁴⁴³⁴ <https://seaborn.pydata.org/>

15.8 Documentazione `ta_sphinx_theme` che utilizza `pyda-`

- Arviz⁴⁴³⁵
- Binder⁴⁴³⁶
- Bokeh⁴⁴³⁷
- CuPy⁴⁴³⁸
- EnOSlib⁴⁴³⁹
- Fairlearn⁴⁴⁴⁰
- Feature-engine⁴⁴⁴¹
- Jupyter⁴⁴⁴²
- Jupyter Book⁴⁴⁴³
- Matplotlib⁴⁴⁴⁴
- MegEngine⁴⁴⁴⁵
- MNE-Python⁴⁴⁴⁶
- NetworkX⁴⁴⁴⁷
- Numpy⁴⁴⁴⁸
- Pandas⁴⁴⁴⁹
- Pystra (continuation of PyRe)⁴⁴⁵⁰
- PyVista⁴⁴⁵¹
- SciPy⁴⁴⁵²

⁴⁴³⁵ <https://python.arviz.org/en/stable/>

⁴⁴³⁶ <https://mybinder.readthedocs.io/en/latest/>

⁴⁴³⁷ <https://docs.bokeh.org/en/latest/>

⁴⁴³⁸ <https://docs.cupy.dev/en/stable/>

⁴⁴³⁹ <https://discovery.gitlabpages.inria.fr/enoslib/>

⁴⁴⁴⁰ <https://fairlearn.org/main/>

⁴⁴⁴¹ <https://feature-engine.readthedocs.io/en/latest/>

⁴⁴⁴² <https://docs.jupyter.org/en/latest/>

⁴⁴⁴³ <https://jupyterbook.org/en/stable/intro.html>

⁴⁴⁴⁴ <https://matplotlib.org/stable/index.html>

⁴⁴⁴⁵ <https://megengine.org.cn/doc/stable/en/>

⁴⁴⁴⁶ <https://mne.tools/stable/>

⁴⁴⁴⁷ <https://networkx.org/documentation/stable/>

⁴⁴⁴⁸ <https://numpy.org/doc/stable/>

⁴⁴⁴⁹ <https://pandas.pydata.org/docs/>

⁴⁴⁵⁰ <https://pystra.github.io/pystra/>

⁴⁴⁵¹ <https://docs.pyvista.org/>

⁴⁴⁵² <https://docs.scipy.org/doc/scipy/>

- [SEPAL](#)⁴⁴⁵³

15.9 Documentazione che utilizza un tema personalizzato o integrato in un sito web

- [AIOHTTP](#)⁴⁴⁵⁴
- [Apache Cassandra](#)⁴⁴⁵⁵
- [Astropy](#)⁴⁴⁵⁶
- [Boto 3](#)⁴⁴⁵⁷
- [CakePHP](#)⁴⁴⁵⁸
- [Ceph](#)⁴⁴⁵⁹
- [Chef](#)⁴⁴⁶⁰
- [CKAN](#)⁴⁴⁶¹
- [Confluent Platform](#)⁴⁴⁶²
- [Django](#)⁴⁴⁶³
- [django CMS](#)⁴⁴⁶⁴
- [Doctrine](#)⁴⁴⁶⁵
- [Enterprise Toolkit for Acrobat products](#)⁴⁴⁶⁶
- [FreeFEM](#)⁴⁴⁶⁷
- [fmt](#)⁴⁴⁶⁸
- [Gameduino](#)⁴⁴⁶⁹
- [gensim](#)⁴⁴⁷⁰

⁴⁴⁵³ <https://docs.sepal.io/en/latest/index.html>

⁴⁴⁵⁴ <https://docs.aiohttp.org/>

⁴⁴⁵⁵ <https://cassandra.apache.org/doc/>

⁴⁴⁵⁶ <https://docs.astropy.org/>

⁴⁴⁵⁷ <https://boto3.readthedocs.io/>

⁴⁴⁵⁸ <https://book.cakephp.org/>

⁴⁴⁵⁹ <https://docs.ceph.com/docs/master/>

⁴⁴⁶⁰ <https://docs.chef.io/>

⁴⁴⁶¹ <https://docs.ckan.org/>

⁴⁴⁶² <https://docs.confluent.io/>

⁴⁴⁶³ <https://docs.djangoproject.com/>

⁴⁴⁶⁴ <https://docs.django-cms.org/>

⁴⁴⁶⁵ <https://www.doctrine-project.org/>

⁴⁴⁶⁶ <https://www.adobe.com/devnet-docs/acrobatetk/>

⁴⁴⁶⁷ <https://doc.freefem.org/introduction/>

⁴⁴⁶⁸ <https://fmt.dev/>

⁴⁴⁶⁹ <https://excamera.com/sphinx/gameduino/>

⁴⁴⁷⁰ <https://radimrehurek.com/gensim/>

- [GeoServer](#)⁴⁴⁷¹
- [gevent](#)⁴⁴⁷²
- [GHC - Glasgow Haskell Compiler](#)⁴⁴⁷³
- [Guzzle](#)⁴⁴⁷⁴
- [H2O.ai](#)⁴⁴⁷⁵
- [Heka](#)⁴⁴⁷⁶
- [Istihza \(Turkish Python documentation project\)](#)⁴⁴⁷⁷
- [JupyterHub](#)⁴⁴⁷⁸
- [Kombu](#)⁴⁴⁷⁹
- [Lasso](#)⁴⁴⁸⁰
- [Mako](#)⁴⁴⁸¹
- [MirrorBrain](#)⁴⁴⁸²
- [Mitiq](#)⁴⁴⁸³
- [MongoDB](#)⁴⁴⁸⁴
- [Music21](#)⁴⁴⁸⁵
- [MyHDL](#)⁴⁴⁸⁶
- [ndnSIM](#)⁴⁴⁸⁷
- [nose](#)⁴⁴⁸⁸
- [ns-3](#)⁴⁴⁸⁹
- [ObjectListView](#)⁴⁴⁹⁰

⁴⁴⁷¹ <https://docs.geoserver.org/>

⁴⁴⁷² <https://www.gevent.org/>

⁴⁴⁷³ <https://downloads.haskell.org/~ghc/master/users-guide/>

⁴⁴⁷⁴ <https://docs.guzzlephp.org/>

⁴⁴⁷⁵ <https://docs.h2o.ai/>

⁴⁴⁷⁶ <https://hekad.readthedocs.io/>

⁴⁴⁷⁷ <https://python-istihza.yazbel.com/>

⁴⁴⁷⁸ <https://jupyterhub.readthedocs.io/>

⁴⁴⁷⁹ <https://kombu.readthedocs.io/>

⁴⁴⁸⁰ <http://www.lassoguide.com/>

⁴⁴⁸¹ <https://docs.makotemplates.org/>

⁴⁴⁸² <https://mirrorbrain-docs.readthedocs.io/>

⁴⁴⁸³ <https://mitiq.readthedocs.io/>

⁴⁴⁸⁴ <https://docs.mongodb.com/>

⁴⁴⁸⁵ <https://web.mit.edu/music21/doc/>

⁴⁴⁸⁶ <https://docs.myhdl.org/>

⁴⁴⁸⁷ <https://ndnsim.net/current/>

⁴⁴⁸⁸ <https://nose.readthedocs.io/>

⁴⁴⁸⁹ <https://www.nsnam.org/documentation/>

⁴⁴⁹⁰ <https://objectlistview.sourceforge.net/python/>

- [OpenERP](#)⁴⁴⁹¹
- [OpenCV](#)⁴⁴⁹²
- [Open Dylan](#)⁴⁴⁹³
- [OpenTURNS](#)⁴⁴⁹⁴
- [Open vSwitch](#)⁴⁴⁹⁵
- [PlatformIO](#)⁴⁴⁹⁶
- [Psycopg](#)⁴⁴⁹⁷
- [PyEphem](#)⁴⁴⁹⁸
- [Pygments](#)⁴⁴⁹⁹
- [Plone User Manual \(German\)](#)⁴⁵⁰⁰
- [PSI4](#)⁴⁵⁰¹
- [PyMOTW](#)⁴⁵⁰²
- [python-aspectlib](#)⁴⁵⁰³ ([sphinx_py3doc_enhanced_theme](#)⁴⁵⁰⁴)
- [QGIS](#)⁴⁵⁰⁵
- [Roundup](#)⁴⁵⁰⁶
- [SaltStack](#)⁴⁵⁰⁷
- [scikit-learn](#)⁴⁵⁰⁸
- [Scrapy](#)⁴⁵⁰⁹
- [Seaborn](#)⁴⁵¹⁰
- [Selenium](#)⁴⁵¹¹

⁴⁴⁹¹ <https://doc.odoo.com/>

⁴⁴⁹² <https://docs.opencv.org/>

⁴⁴⁹³ <https://opendylan.org/>

⁴⁴⁹⁴ <https://openturns.github.io/openturns/latest/>

⁴⁴⁹⁵ <https://docs.openvswitch.org/>

⁴⁴⁹⁶ <https://docs.platformio.org/>

⁴⁴⁹⁷ <https://www.psycopg.org/docs/>

⁴⁴⁹⁸ <https://rhodesmill.org/pyephem/>

⁴⁴⁹⁹ <https://pygments.org/docs/>

⁴⁵⁰⁰ <https://www.hasecke.com/plone-benutzerhandbuch/4.0/>

⁴⁵⁰¹ <https://www.psicode.org/psi4manual/master/index.html>

⁴⁵⁰² <https://pymotw.com/2/>

⁴⁵⁰³ <https://python-aspectlib.readthedocs.io/>

⁴⁵⁰⁴ https://pypi.org/project/sphinx_py3doc_enhanced_theme/

⁴⁵⁰⁵ <https://qgis.org/en/docs/index.html>

⁴⁵⁰⁶ <https://www.roundup-tracker.org/>

⁴⁵⁰⁷ <https://docs.saltstack.com/>

⁴⁵⁰⁸ <https://scikit-learn.org/stable/>

⁴⁵⁰⁹ <https://doc.scrapy.org/>

⁴⁵¹⁰ <https://seaborn.pydata.org/>

⁴⁵¹¹ <https://docs.seleniumhq.org/docs/>

- Self⁴⁵¹²
- Substance D⁴⁵¹³
- Sulu⁴⁵¹⁴
- SQLAlchemy⁴⁵¹⁵
- tinyTiM⁴⁵¹⁶
- Twisted⁴⁵¹⁷
- Ubuntu Packaging Guide⁴⁵¹⁸
- WTForms⁴⁵¹⁹

15.10 Home page e altri siti non di documentazione

- Alan Crosswell's Using the Django REST Framework and DRF-JSONAPI⁴⁵²⁰
- Arizona State University PHY494/PHY598/CHM598 Simulation approaches to Bio-and Nanophysics⁴⁵²¹ (classic)
- Benoit Boissinot⁴⁵²² (classic, customized)
- EBI Cloud Consultancy Team⁴⁵²³ (sphinx_rtd_theme)
- Eric Holscher⁴⁵²⁴ (alabaster)
- Florian Diesch⁴⁵²⁵
- Institute for the Design of Advanced Energy Systems (IDAES)⁴⁵²⁶ (sphinx_rtd_theme)
- IDAES Examples⁴⁵²⁷ (sphinx_rtd_theme)
- Lei Ma's Statistical Mechanics lecture notes⁴⁵²⁸ (sphinx_bootstrap_theme)
- PyXLL⁴⁵²⁹ (sphinx_bootstrap_theme, customized)

⁴⁵¹² <https://www.selflanguage.org/>

⁴⁵¹³ <https://docs.pylonsproject.org/projects/substanced/>

⁴⁵¹⁴ <https://docs.sulu.io/>

⁴⁵¹⁵ <https://docs.sqlalchemy.org/>

⁴⁵¹⁶ <https://tinytim.sourceforge.net/docs/2.0/>

⁴⁵¹⁷ <https://twistedmatrix.com/documents/current/>

⁴⁵¹⁸ <https://packaging.ubuntu.com/html/>

⁴⁵¹⁹ <https://wtforms.readthedocs.io/>

⁴⁵²⁰ <https://www.columbia.edu/~alan/django-jsonapi-training/>

⁴⁵²¹ <https://becksteinlab.physics.asu.edu/pages/courses/2013/SimBioNano/>

⁴⁵²² <https://bboissin.appspot.com/>

⁴⁵²³ <https://tsi-ccdod.readthedocs.io/>

⁴⁵²⁴ <https://ericholscher.com/>

⁴⁵²⁵ <https://www.florian-diesch.de/>

⁴⁵²⁶ <https://idaes-pse.readthedocs.io/>

⁴⁵²⁷ <https://idaes.github.io/examples-pse/>

⁴⁵²⁸ <https://statisticalphysics.leima.is/>

⁴⁵²⁹ <https://www.pyxll.com/>

- SciPy Cookbook⁴⁵³⁰ (sphinx_rtd_theme)
- Tech writer at work blog⁴⁵³¹ (custom theme)
- UC Berkeley ME233 Advanced Control Systems II course⁴⁵³² (sphinxdoc)
- Željko Svedružić's Biomolecular Structure and Function Laboratory (BioSFLab)⁴⁵³³ (sphinx_bootstrap_theme)

15.11 Libri prodotti utilizzando Sphinx

- "The Art of Community" (Japanese translation)⁴⁵³⁴
- "Die Wahrheit des Sehens. Der DEKALOG von Krzysztof Kieślowski"⁴⁵³⁵
- "Expert Python Programming"⁴⁵³⁶
- "Expert Python Programming" (Japanese translation)⁴⁵³⁷
- "Expert Python Programming 2nd Edition" (Japanese translation)⁴⁵³⁸
- "The Hitchhiker's Guide to Python"⁴⁵³⁹
- "LassoGuide"⁴⁵⁴⁰
- "Learning Sphinx" (in Japanese)⁴⁵⁴¹
- "Learning System Programming with Go (Japanese)"⁴⁵⁴²
- "Mercurial: the definitive guide (Second edition)"⁴⁵⁴³
- "Mithril -- The fastest clientside MVC (Japanese)"⁴⁵⁴⁴
- "Pioneers and Prominent Men of Utah"
- "Pomodoro Technique Illustrated" (Japanese translation)⁴⁵⁴⁵
- "Professional Software Development"⁴⁵⁴⁶

⁴⁵³⁰ <https://scipy-cookbook.readthedocs.io/>

⁴⁵³¹ <https://documatt.com/blog/>

⁴⁵³² <https://berkeley-me233.github.io/>

⁴⁵³³ <https://svedruziclab.github.io/>

⁴⁵³⁴ <https://www.oreilly.co.jp/books/9784873114958/>

⁴⁵³⁵ <https://literatur.hasecke.com/post/die-wahrheit-des-sehens-dekalog-kieslowski/>

⁴⁵³⁶ <https://www.packtpub.com/application-development/expert-python-programming>

⁴⁵³⁷ <https://www.amazon.co.jp/dp/4048686291/>

⁴⁵³⁸ <https://www.amazon.co.jp/dp/4048930613/>

⁴⁵³⁹ <https://docs.python-guide.org/>

⁴⁵⁴⁰ <http://www.lassoguide.com/>

⁴⁵⁴¹ <https://www.oreilly.co.jp/books/9784873116488/>

⁴⁵⁴² <https://www.lambdanote.com/products/go>

⁴⁵⁴³ <https://book.mercurial-scm.org/>

⁴⁵⁴⁴ <https://www.oreilly.co.jp/books/9784873117447/>

⁴⁵⁴⁵ <https://www.amazon.co.jp/dp/4048689525/>

⁴⁵⁴⁶ https://mixmastamyk.bitbucket.io/pro_soft_dev/

- "Python Professional Programming" (in Japanese)⁴⁵⁴⁷
- "Python Professional Programming 2nd Edition" (in Japanese)⁴⁵⁴⁸
- "Python Professional Programming 3rd Edition" (in Japanese)⁴⁵⁴⁹
- Python Course by Yuri Petrov (Russian)⁴⁵⁵⁰
- "Real World HTTP -- Learning The Internet and Web Technology via its history and code (Japanese)"⁴⁵⁵¹
- "Redmine Primer 5th Edition (in Japanese)"⁴⁵⁵²
- "The repoze.bfg Web Application Framework"⁴⁵⁵³
- "The Self-Taught Programmer" (Japanese translation)⁴⁵⁵⁴
- "Simple and Steady Way of Learning for Software Engineering" (in Japanese)⁴⁵⁵⁵
- "Software-Dokumentation mit Sphinx"⁴⁵⁵⁶
- "Theoretical Physics Reference"⁴⁵⁵⁷
- "The Varnish Book"⁴⁵⁵⁸

15.12 Tesi prodotte utilizzando Sphinx

- "Content Conditioning and Distribution for Dynamic Virtual Worlds"⁴⁵⁵⁹
- "The Sphinx Thesis Resource"⁴⁵⁶⁰

⁴⁵⁴⁷ <https://www.amazon.co.jp/dp/4798032948/>

⁴⁵⁴⁸ <https://www.amazon.co.jp/dp/479804315X/>

⁴⁵⁴⁹ <https://www.amazon.co.jp/dp/4798053821/>

⁴⁵⁵⁰ <https://www.yuripetrov.ru/edu/python>

⁴⁵⁵¹ <https://www.oreilly.co.jp/books/9784873118048/>

⁴⁵⁵² <https://www.shuwasystem.co.jp/products/7980html/4825.html>

⁴⁵⁵³ <https://www.amazon.com/repoze-bfg-Web-Application-Framework-Version/dp/0615345379>

⁴⁵⁵⁴ <https://www.amazon.co.jp/dp/4822292274/>

⁴⁵⁵⁵ <https://www.amazon.co.jp/dp/477414259X/>

⁴⁵⁵⁶ <https://www.amazon.de/dp/1497448689/>

⁴⁵⁵⁷ <https://www.theoretical-physics.net/>

⁴⁵⁵⁸ <https://info.varnish-software.com/the-varnish-book>

⁴⁵⁵⁹ <https://www.cs.princeton.edu/research/techreps/TR-941-12>

⁴⁵⁶⁰ <https://jterrace.github.io/sphinxtr/>

15.13 Progetti che integrano la funzionalità di Sphinx

- [Read the Docs](https://readthedocs.org/)⁴⁵⁶¹, una piattaforma di hosting di documentazione software-as-a-service, utilizza Sphinx per creare automaticamente aggiornamenti della documentazione che vengono inviati a GitHub.
- [Spyder](https://docs.spyder-ide.org/current/panes/help.html)⁴⁵⁶², the Scientific Python Development Environment, utilizza Sphinx nel suo riquadro della guida per visualizzare automaticamente una ricca documentazione per funzioni, classi e metodi o on-demand.

⁴⁵⁶¹ <https://readthedocs.org/>

⁴⁵⁶² <https://docs.spyder-ide.org/current/panes/help.html>

a

`sphinx.addnodes`, 410
`sphinx.application`, 357

b

`sphinx.builders`, 152
`sphinx.builders.changes`, 161
`sphinx.builders.dirhtml`, 153
`sphinx.builders.dummy`, 162
`sphinx.builders.epub3`, 156
`sphinx.builders.gettext`, 161
`sphinx.builders.html`, 153
`sphinx.builders.latex`, 156
`sphinx.builders.linkcheck`, 162
`sphinx.builders.manpage`, 158
`sphinx.builders.singlehtml`, 154
`sphinx.builders.texinfo`, 159
`sphinx.builders.text`, 158
`sphinx.builders.xml`, 162

C

`conf`, 91

d

`docutils.parsers.rst`, 396
`sphinx.directives`, 405
`sphinx.domains`, 399
`sphinx.domains.python`, 408

e

`sphinx.environment`, 388
`sphinx.environment.collectors`, 393
`sphinx.errors`, 377
`sphinx.ext.autodoc`, 206
`sphinx.ext.autosectionlabel`, 223
`sphinx.ext.autosummary`, 224

`sphinx.ext.coverage`, 229
`sphinx.ext.doctest`, 231
`sphinx.ext.duration`, 238
`sphinx.ext.extlinks`, 238
`sphinx.ext.githubpages`, 239
`sphinx.ext.graphviz`, 240
`sphinx.ext.ifconfig`, 243
`sphinx.ext.imgconverter`, 244
`sphinx.ext.imgmath`, 254
`sphinx.ext.inheritance_diagram`, 245
`sphinx.ext.intersphinx`, 248
`sphinx.ext.jsmath`, 258
`sphinx.ext.linkcode`, 253
`sphinx.ext.mathbase`, 254
`sphinx.ext.mathjax`, 256
`sphinx.ext.napoleon`, 259
`sphinx.ext.todo`, 270
`sphinx.ext.viewcode`, 271

I

`latex`, 451

p

`sphinx.parsers`, 409

S

`sphinxcontrib.applehelp`, 155
`sphinxcontrib.devhelp`, 155
`sphinxcontrib.htmlhelp`, 154
`sphinxcontrib.qthelp`, 154

Indice
