

Org Mode Compact Guide

Release 9.7

The Org Mode Developers Luca Canali

June 22, 2024

Contents

1	Copying	3
2	Introduzione	3
3	Struttura del documento	4
3.1	Intestazioni	5
3.2	Ciclo di visualizzazione	5
3.3	Movimento	6
3.4	Modificare la struttura	6
3.5	Alberi sparsi	7
3.6	Liste semplici	8
4	Tabelle	9
5	Collegamenti ipertestuali	11
6	Elementi TODO	13
6.1	TODO Basics	14
6.2	Workflow multi-stato	14
6.3	Registrazioni dei progressi	15
6.4	Priorità	17
6.5	Breaking Down Tasks	17
6.6	Liste selezionabili	17
7	TAG	18
8	Proprietà	21

9	Data e orario	22
9.1	Timestamp	22
9.2	Creare un timestamp	24
9.3	Scadenze e schedulazioni	25
9.4	Cronometra il tempo di lavoro	26
10	Cattura, sposta e archivia	26
10.1	Cattura	27
10.2	Sposta e Copia	28
10.3	Archiviare	29
11	Visualizzazioni Agenda	29
11.1	File agenda	30
11.2	Agenda Dispatcher	30
11.3	Built-in Agenda Views	31
11.4	Global TODO List	31
11.5	Trova i tag e le proprietà	32
11.6	Vista di ricerca	32
11.7	Agenda Commands	33
11.8	Personalizzare la vista agenda	35
12	Markup	36
12.1	Paragrafi	36
12.2	Enfasi e mono-space	37
12.3	L ^A T _E X integrato	37
12.4	Esempi letterali	38
12.5	Immagini	38
12.6	Creare note a piè di pagina	39
13	Esportazione	39
13.1	Il dispatcher di esportazione	39
13.2	Impostazioni di esportazione	40
13.3	Tabelle dei contenuti	40
13.4	Includere file	40
13.5	Linee di commento	41
13.6	Esportazione ASCII/UTF-8	41
13.7	Esportazione HTML	41
13.8	Esportazione L ^A T _E X	42
13.9	Esportazione iCalendar	43

14 Pubblicare	43
15 Lavorare con il codice sorgente	44
16 Miscellanee	48
17 Traduzione	49

1 Copying

Copyright © 2004–2024 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with the Front-Cover Texts being “A GNU Manual,” and with the Back-Cover Texts as in (a) below. A copy of the license is included in the section entitled “GNU Free Documentation License.” in the full Org manual, which is distributed together with this compact guide.

(a) The FSF’s Back-Cover Text is: “You have the freedom to copy and modify this GNU manual.”

2 Introduzione

Org é una modalità per prendere note, mantenere liste TODO ed a pianificare progetti con un sistema di testo semplice veloce ed efficace. È anche un sistema di creazione e pubblicazione che supporta il lavoro con i blocchi di codice per la programmazione letterale e la ricerca riproducibile.

Questo documento é un derivato molto compresso dal più completo manuale Org mode. Contiene tutte le funzionalità ed i comandi di base, assieme ad importanti suggerimenti per la personalizzazione. È destinato ai principianti che potrebbero tirarsi indietro, a causa delle dimensioni, davanti ad un manuale di 200 pagine.

Installazione

Se usi una versione di Org che fa parte della distribuzione di Emacs, per favore salta questa sezione e vai direttamente all’Attivazione.

Se hai scaricato Org dal web, solitamente viene distribuito come file `.zip`, `.tar` oppure come archivio Git, é meglio avviarlo direttamente dalla directory di distribuzione. Hai bisogno di aggiungere la sottodirectory `lisp/` alla path di Emacs. Per fare questo aggiungi la seguente linea nel tuo file di inizializzazione:

```
(add-to-list 'load-path "~/path/to/orgdir/lisp")
```

Se hai usato git o tar ball per ottenere Org, hai bisogno di avviare il seguente comando per generare le informazioni di autoload.

```
make autoload
```

Attivazione

Aggiungi le seguenti linee al tuo file di init per definire delle chiavi *globali* per tre comandi che sono utili in ogni buffer Emacs, non solo nel buffer Org. Per favore scegli le chiavi più comode per te.

```
(global-set-key (kbd "C-c l") #'org-store-link)
(global-set-key (kbd "C-c a") #'org-agenda)
(global-set-key (kbd "C-c c") #'org-capture)
```

I file con estensione `.org` sono posti in modo automatico nella modalità Org.

Feedback

Se riscontri dei problemi con Org o se hai delle domande, osservazioni o idee a riguardo, per favore invia una mail alla mailing list <mailto:emacs-orgmode@gnu.org>. Per informazioni su come inviare dei report sui bug, guarda il [\[\[info:org\]\]](#) manuale principale.

3 Struttura del documento

Org é uno schematizzato-re. La schematizzazione permette al documento di essere organizzato in strutture gerarchiche, che, almeno per me, sono la migliore rappresentazione delle note e dei pensieri. Un'osservazione di questa struttura si ottiene piegando, es. nascondendo larghe porzioni del documento per mostrarne solo la struttura generale e la parte dove al momento si sta lavorando. Org semplifica molto bene l'uso delle strutture comprimendo l'intera vista e nascondendo le funzionalità con un singolo comando `org-cycle` che é legato al tasto TAB.

3.1 Intestazioni

Le intestazioni definiscono la struttura di uno schema ad albero. Le intestazioni in Org iniziano dal margine sinistro¹ con uno o più asterischi seguiti da uno spazio. Per esempio:

```
* Livello iniziale dei titoli
** Secondo livello
*** Terzo livello
Un po di testo
*** Terzo livello
Altro testo
* Un'altro livello iniziale
```

Nota che un'intestazione nominata dopo `org-footnote-section`, che di default é detto `Footnotes`, é considerato come speciale. Un sotto-albero con questa intestazione sarà ignorata silenziosamente dalla funzione di esportazione.

Alcune persone trovano i vari asterischi troppo fastidiosi e preferiscono uno schema con degli spazi vuoti seguiti da un singolo asterisco. Guarda nelle 16 per un setup che lo realizza.

3.2 Ciclo di visualizzazione

Gli schemi rendono possibile nascondere parte del testo del buffer. Org usa giusto due comandi, legati a TAB e S-TAB (`org-cycle` e `org-shifttab`) per scambiare la visibilità nel buffer.

TAB (`org-cycle`) *Ciclo del sotto-albero*: Ruota il sotto-albero corrente attraverso gli stai

```
,-> CHIUSO -> FIGLI -> SOTTO-ALBERO --.
'-----'
```

Quando viene chiamato con l'argomento prefisso (C-u TAB) o con il tasto Shiftf, viene invocato il ciclo globale.

S-TAB (`org-global-cycle`), **C-u TAB** (`org-cycle`) *Ciclo globale*: Ruota l'intero buffer attraverso gli stati

¹Guarda la variabile `org-special-ctrl-a/e` per configurare speciali comportamenti di C-a e C-e nei titoli.

```
, -> OVERVIEW -> CONTENUTO -> MOSTRA TUTTO --.
,-----,
```

C-u C-u C-u TAB (org-show-all) Mostra tutto incluso i drawer.

Quando Emacs visita per la prima volta un file Org, lo stato globale é settato a OVERVIEW, es. solo i livelli più alti delle intestazione sono visibili. Questo può essere configurato tramite la variabile `org-startup-folder` o nell'intestazione del file aggiungendo una keyword `STARTUP` con `overview`, `content`, `shoall`, `showeverything` o `show<n>levels` ($n = 2..5$) tipo:

```
#+STARTUP: content
```

3.3 Movimento

I comandi seguenti ti fanno saltare ad altre intestazioni nel buffer

C-c C-n (org-next-visible-heading) Intestazione successivo

C-c C-p (org-previous-visible-heading) Intestazione precedente

C-c C-f (org-backward-heading-same-level) Intestazione successivo nello stesso livello

C-c C-b (outline-backward-same-level) Intestazione precedente nello stesso livello

C-c C-u (outline-up-heading) Torna all'intestazione più in alto

3.4 Modificare la struttura

M-RET (org-meta-return) Inserisce un nuovo titolo con lo stesso livello del corrente. Se il punto é una lista di oggetti, viene creato un nuovo oggetto (vedere 3.6). Quando questo comando viene usato nel mezzo di una linea, la linea viene divisa e il resto della linea diventa la nuova intestazione²

M-S-RET (org-inser-todo-heading) Inserisce un nuovo oggetto TODO con lo stesso livello dell'intestazione corrente

²Se non vuoi che la linea sia divisa, personalizza la variabile `org-M-RET-may-split-line`

TAB (org-cycle) in un oggetto vuoto In un nuovo oggetto che non ha ancora del testo, il TAB cicla tra i vari livelli

M-SINISTRA (org-metaleft), M-DESTRA (org-metaright) Promuove o retrocede l'intestazione corrente di un livello

M-SU (org-move-subtree-up), M-GIÙ (org-move-subtree-down) Muove il sotto-albero su o giù es. scambia con il precedente o il successivo sotto-albero nello stesso livello.

C-c C-w (org-refile) Ricolloca un oggetto o una regione in una posizione differente. Vedere 10.2.

C-x n s (org-narrow-to-subtree), C-x n w (widen) Allarga il buffer al corrente sotto-albero e lo nasconde nuovamente

Quando c'è una regione attiva (Modalità Contrassegno di transito), promuove o retrocede lavorando su tutti i titoli nella regione.

3.5 Alberi sparsi

Una funzionalità importante della modalità Org è l'abilità di costruire *alberi sparsi* per le informazioni selezionate in una struttura ad albero, in modo che l'intero documento sia piegato il più possibile, ma le informazioni selezionate siano comunque visibili assieme alla struttura dell'intestazione sopra di esso³. Provalo subito e vedrai immediatamente come lavora.

La modalità Org contiene diversi comandi che creano questi alberi, a tutti questi comandi si può accedere attraverso uno dispatcher.

C-c / (org-sparse-tree) Mostra un prompt in cui selezionare una chiave per creare un comando di sparse-tree

C-c / r (org-occur) Mostra un prompt per un'espressione regolare (reg-exp) e mostra un albero sparso con tutte le occorrenze trovate. Ogni occorrenza viene sottolineata; La sottolineatura svanisce se si preme C-c C-c.

L'altro comando di sparse tree seleziona le intestazioni basandosi sulle chiavi dei TODO, tags o proprietà che verrà discusso più avanti in questo documento.

³Vedi anche la variabile `org-show-context-detail` per decidere quanto contesto mostrare attorno ad ogni selezione.

3.6 Liste semplici

Assieme gli oggetti della struttura ad albero, gli elenchi formattati manualmente possono fornire una struttura aggiuntiva. Inoltre forniscono un modo per creare liste selezionabili o caselle di controllo (vedi 6.6). Org supporta la modifica di queste liste ed ogni esportatore (Vedi 13) può analizzarlo e formattarlo.

Org riconosce le liste ordinate, non ordinate e descritte.

- Gli elementi di una lista *non ordinata* iniziano con un -, + o un * come puntatore.
- Gli elementi di una lista *ordinata* iniziano con un 1. o 1).
- Le liste *descritte* usano :: per separare il *termine* dalla descrizione.

Gli elementi appartenenti allo stesso elenco devono avere la stessa indentazione sulla prima riga. Un elemento termina prima della riga successiva che é rientrato come il su elenco/numero o meno. Un elenco termina quanto tutti gli elementi vengono chiusi o dopo due righe vuote.

* Lord of the Rings

My favorite scenes are (in this order)

1. The attack of the Rohirrim

2. Eowyn's fight with the witch king

+ this was already my favorite scene in the book

+ I really like Miranda Otto.

Important actors in this film are:

- Elijah Wood :: He plays Frodo

- Sean Astin :: He plays Sam, Frodo's friend.

I seguenti comandi agiscono sugli elementi quando il punto si trova nella prima riga di un elemento (la riga con il punto elenco o il numero).

TAB (`org-cycle`) Gli elementi possono essere ripiegati come le intestazioni.

M-RET (`org-insert-heading`) Inserisce un nuovo elemento nel livello corrente. Con un argomento prefisso, forza una nuova intestazione (vedere 3.4).

M-S-RET (`org-insert-todo-heading`) Inserisce un nuovo elemento con un casella di controllo (vedere 6.6).

M-S-SU (`org-move-item-up`), **M-S-GIU** (`org-move-item-down`) Muove l'elemento, inclusi i sotto-elementi, su/giù (spostandoli con il precedente/successivo elemento nella medesima indentazione). Se la lista é ordinata, la rinumerata in automatico.

M-SINISTRA (`org-do-promote`), **M-DESTRA** (`org-do-subtree`) Aumenta/diminuisce l'indentazione dell'elemento, fatta esclusione per i figli.

M-S-SINISTRA (`org-promote-subtree`), **M-S-DESTRA** (`org-demote-subtree`) Aumenta/diminuisce l'indentazione dell'elemento, esclusi i figli.

C-c C-c (`org-toggle-checkbox`) Se c'è una casella di controllo (vedere 6.6) nella linea degli elementi, cambia lo stato della checkbox. Verificando anche i puntatori e la consistenza dell'indentazione in tutta la lista.

C-c - (`org-cycle-list-bullet`) Scorre tra la lista dei livelli degli elementi attraverso differenti tipi di puntatori (-, +, *, 1., 1)).

4 Tabelle

Org ha un editor di tabelle veloce ed intuitivo. Come per i fogli di calcolo sono supportati i calcoli, in connessione con il pacchetto Emacs Calc (vedere GNU Emacs Calculator Manuale).

Org rende facile formattare le tabelle in semplice codice ASCII. Ogni linea con | come primo carattere senza spazi viene considerato come parte di una tabella. Il carattere | é anche un separatore di colonne. Una tabella si presenta in questo modo:

```
| Name | Phone | Age |
|-----+-----+-----|
| Peter | 1234 | 17 |
| Anna | 43212 | 25 |
```

Una tabella viene riallineata automaticamente ogni volta che viene premuto TAB o RET oppure C-c C-c quando ci si trova all'interno della tabella. Il TAB inoltre muove il punto al campo successivo (RET alla riga successiva) e crea nuove righe alla fine della tabella o prima delle linee orizzontali. L'indentation di una tabella é settata nella prima linea. Ogni linea che inizia con un | - é considerata un separatore orizzontale e viene espanso al successivo riallineamento per occupare tutta la larghezza della tabella. Quindi per creare la tabella precedente devi solo digitare

```
|Name|Phone|Age|  
|-
```

e premere TAB per allineare la tabella ed iniziare a riempire i campi. Per voler essere più veloce digita `|Name|Phone|Age` seguito da C-c RET.

Quando digiti un testo in un campo, Org tratta DEL, Backspace e tutti gli altri caratteri, in un modo speciale, in modo che l'inserimento e la cancellazione non sposti gli altri campi. Inoltre quando digiti, *subito dopo che il punto viene mosso in un nuovo campo con TAB, S-TAB o RET*, il campo viene automaticamente creato bianco.

Creazione e conversione

C-c | (`org-table-create-or-convert-from-region`) Converte la regione attiva in una tabella. Se ogni linea contiene almeno un carattere TAB, la funzione assume che le cose siano separate da tab. Se ogni linea é separata da virgola (CSV) assume che sia un file separato da virgole. Altrimenti, le linee sono divise in capi tramite gli spazi bianchi.

Se non ci sono regioni attive, questo comando crea una tabella Org vuota. Ma é più facile inserendo `| N a m e | P h o n e | A g e` RET `| -` TAB.

Ri-allineamento e spostamento dei campi

C-c C-c (`org-table-align`) Ri-allinea la tabella senza spostare il punto.

TAB (`org-table-next-field`) Ri-allinea la tabella, muovendosi nel campo successivo. Crea una nuova riga se necessario.

S-TAB (`org-table-previous-field`) Ri-allinea la tabella, muovendo nel campo precedente.

RET (`org-table-next-row`) Ri-allinea la tabella e muove in basso nella riga successiva. Creando una nuova riga se necessario.

S-SU (`org-table-move-cell-up`), **S-GIU** (`org-table-move-cell-down`), **S-SINISTRA** (`org-table-move-cell-left`), **S-DESTRA** (`org-table-move-cell-right`) Muove le celle su, giù, a sinistra e a destra spostando la cella adiacente.

Modifica colonne e righe

M-SINISTRA (`org-table-move-column-left`), **M-DESTRA** (`org-table-move-column-right`) Muove la colonna corrente a sinistra/destra.

M-S-SINISTRA (`org-table-delete-column`) Elimina la colonna corrente.

M-S-DESTRA (`org-table-insert-column`) Inserisce una nuova colonna a sinistra della posizione del punto.

M-SU (`org-tablet-move-row`), **M-GIU** (`org-tablet-move-row-down`) Muove la riga corrente su/giù.

M-S-SU (`org-table-kill-row`) Elimina la riga corrente o la linea orizzontale.

M-S-GIU (`org-table-insert-row`) Inserisce una nuova riga sopra la corrente. Con un argomento prefisso la linea viene creata sotto la corrente.

C-c - (`org-table-insert-hline`) Inserisce una linea sotto la corrente. Con un argomento prefisso la linea viene creata sopra la corrente.

C-c RET (`org-table-hline-and-move`) Inserisce una linea sotto la corrente e muove il punto nella riga sotto questa linea.

C-c ^ (`org-table-sort-lines`) Ordina le linee della tabella in regioni. La posizione del punto indica la colonna da usare per l'ordinamento ed il range delle linee é il range delle linee tra le linee orizzontali più vicine o l'intera tabella.

5 Collegamenti ipertestuali

Come nell'HTML, Org fornisce i link all'interno di un file, link esterni ad altri file, articoli Usenet, email e molto altro.

Org riconosce i semplici URIs, é possibile racchiuderli tra parentesi quadrate ed attivarli come link cliccabili. Il formato generale dei link si presenta simile a questo:

```
[[LINK] [DESCRIZIONE]]
```

od in alternativa

```
[[LINK]]
```

Quando un link nel buffer é completo, con tutte le parentesi, Org cambia la sua visualizzazione con DESCRIZIONE al posto di `[[LINK] [DESCRIZIONE]]` e in LINK al posto di `[[LINK]]`. Per modificare la parte invisibile del LINK usa C-c C-l con il punto sul link.

Link interni

Se i link non sono rappresentati come URI, vengono considerati come interni al file corrente. Il caso più rilevante é un link tipo `[[#my-custom-id]]` che viene collegato ad un elemento con la proprietà `CUSTOM_ID` settata a `my-custom-id`.

Link tipo `[[My Target]]` o `[[My Targhet][Find my target]]` cerca, nel file corrente, un testo che corrisponda al targhet `<<My Target>>`.

Link esterni

Org supporta i link ai file, siti web, Usenet, messaggi email, elementi BBDB database e sia le conversazioni che i logs IRC. I collegamenti sono simili agli URL. Iniziano con una breve stringa di identificazione seguita dai due-punti `:`. Non ci sono spazi dopo i due-punti. Ecco alcuni esempi:

<code>http://www.astro.uva.nl/=dominikper</code>	Nel web
<code>file:/home/dominik/images/jupiter.jpg</code>	file, path assoluta
<code>/home/dominik/images/jupiter.jpg</code>	come sopra
<code>file:papers/last.pdf</code>	file, path relativa
<code>./papaers/last.pdf</code>	come sopra
<code>file:project.org</code>	un'altro file org
<code>docview:papers/last.pdf::NNN</code>	apre in modalità DocView alla pagina NNN
<code>id:B7423F4D-2E8A-471B-8810-C40F074717E9</code>	link ad un titolo con ID
<code>news:comp.emacs</code>	Usenet link
<code>mailto:adent@galaxy.net</code>	link mail
<code>mhe:folder#id</code>	link messaggio MH-E
<code>rmail:folder#id</code>	link messaggio Rmail
<code>gnus:gruop#id</code>	link articolo Gnus
<code>bbdb:R.*Stalman</code>	link BBDB (con regexp)
<code>irc:/irc.com/#emacs/bob</code>	link IRC
<code>info:org#Hyperlink</code>	link nodo Info

I collegamento ai file possono contenere informazioni addizionali per consentire ad Emacs di saltare ad una particolare posizione nel file quando si segue un link. Questa può essere un numero di linea o un opzione di ricerca dopo i due due-punti. Di seguito alcuni esempi, assieme alle spiegazioni:

<code>file:~/code/main.c::255</code>	Trova la linea 255
<code>file:~/xx.org::My Target</code>	Trova <code><<My Target>></code>
<code>[[file:~/xx.org::#my-custom-id]]</code>	Trova l'elemento con l'ID personalizzato

Manipolare i collegamenti

Org fornisce un modo per creare link con la sintassi corretta, per inserirla in un file Org e per seguire il link.

La funzione principale è **org-store-link** richiamabile con M-x org-store-link. Per la sua importanza, suggeriamo di abbinarlo ad una combinazione più comoda (vedere Attivazione). Salva un link nella posizione attuale. Il link viene salvato per essere successivamente inserito in un buffer Org – vedi sotto.

Da un buffer Org, i comandi seguenti creano, navigano o, più generalmente, agiscono sui link.

C-c C-l (org-insert-link) Inserisce un link. Questo richiede un link che sarà inserito nel buffer. Puoi inserire il link o usare la history SU e GIU per accedere ai link salvati. Richiederà la parte di descrizione per il link.

Quando chiamato con l'argomento prefisso C-u il completamento dei nomi viene usato per l'inserimento di un nome di file.

C-c C-l (Che punta ad un link esistente) (org-insert-link) Quando il punto è su un link esistente, C-c C-l permette la modifica del link e della parte della descrizione.

C-c C-o (open-link-at-point) Apre il link su cui si si trova.

C-c & (org-mark-ring-goto) Torna indietro a una posizione registrata. Una posizione è registrata tramite il comando di link interno e tramite C-c %. Usando questo comando più volte in diretta successione si muove attraverso un anello di posizioni precedenti.

6 Elementi TODO

Org mode non necessita che le liste TODO risiedano in documenti separati. Tant'è che gli elementi TODO sono parte del file di note, perché gli elementi TODO vengono solitamente usati quando si prendono note! Con la modalità Org, semplicemente si marca un oggetto in un albero mettendo all'inizio un elemento TODO. In questo modo, le informazioni non vengono duplicate e gli elementi TODO rimangono nel contesto in modo che emergano.

La modalità Org ti fornisce dei metodi per una visione d'insieme delle cose che devi fare, raccogliendoli da diversi file.

6.1 Funzionalità base dei TODO

Un'intestazione è un elemento TODO quando inizia con la parola `TODO`, per esempio:

```
*** TODO Write letter to Sam Fortune
```

I comandi più importanti quando si lavora con i TODO sono:

C-c C-t (org-todo) Cambia lo stato del TODO dell'elemento corrente

```
,-> (non segnato) -> TODO -> DONE --.  
'-----'
```

La stessa rotazione può anche essere fatta “da remoto”, dal buffer dell'agenda, tramite la chiave `t` (vedi 11.7).

S-RIGHT (org-shiftright), S-LEFT (org-shiftright) Seleziona il seguente/precedente stato TODO, similmente al ciclo.

C-c / t (org-show-todo-tree) Visualizza gli elementi TODO in un *albero sparso* (vedi 3.5). Piega l'intero buffer ma mostra tutti gli elementi TODO - con uno stato non-DONE - e la gerarchia delle intestazioni sopra di loro.

M-x org-agenda t (org-todo-list) Mostra la lista globale dei TODO. Colleziona tutti gli elementi TODO (con stato non-DONE) da tutti i file agenda (vedi 11) in un unico buffer. Vedi 11.4 per maggiori informazioni.

S-M-RET (org-insert-todo-heading) Inserisce un nuovo oggetto TODO sotto al corrente.

Cambiando lo stato di un TODO può anche cambiare i tag. Vedi la docstring dell'opzione `org-todo-state-tags-triggers` per dettagli.

6.2 Workflow multi-stato

Puoi usare le parole chiave TODO per indicare uno stato di lavoro *sequenziale*:

```
(setq org-todo-keywords  
  '((sequence "TODO" "FEEDBACK" "VERIFY" "|" "DONE" "DELEGATED")))
```

La barra verticale separa le keyword `TODO` (stati che *necessitano azioni*) da quelle `DONE` (che *non necessitano future azioni*). Se non fornisci una barra di separazione, l'ultimo stato viene usato come stato `DONE`. Con questo setup, il comando `C-c C-t` cicla un oggetto da `TODO` a `FEEDBACK` quindi `VERIFY` ed in fine `DONE` e `DELEGATED`.

A volte vorresti usare, contemporaneamente, diversi insiemi di parole chiave `TODO`. Per esempio, potresti voler avere il basico `TODO` `!=DONE`, ma anche un workflow per il bug fix. Il tuo setup potrebbe essere simile a questo:

```
(setq org-todo-keywords
  '(sequence "TODO(t)" "|" "DONE(d)")
    (sequence "REPORT(r)" "BUG(b)" "KNOWNCAUSE(k)" "|" "FIXED(f)")))
```

Le keyword dovrebbero essere tutte differenti, questo aiuta Org mode a tenere traccia di quale sotto-sequenza debba essere usata dall'oggetto dato. L'esempio mostra, anche, come definire delle chiavi per un accesso più veloce ad un particolare stato, aggiungendo una lettera tra parentesi dopo ogni parola chiave - riceverai un prompt dopo aver premuto `C-c C-t`.

Per definire delle keyword `TODO` che siano valide solo in un singolo file, usa il seguente testo in una qualsiasi parte del file.

```
#+TODO: TODO(t) | DONE(d)
#+TODO: REPORT(r) BUG(b) KNOWNCAUSE(k) | FIXED(f)
#+TODO: | CANCELED(c)
```

Dopo aver cambiato una di queste linee, usa `C-c C-c` con il cursore nella linea dove hai fatto il cambiamento per farlo conoscere alla modalità Org.

6.3 Registrazioni dei progressi

Per registrare un timestamp e una nota quando cambi lo stato di un `TODO`, esegui il comando `org-todo` con un argomento prefisso.

C-u C-c C-t (org-todo) Fornisce un prompt per una nota e registra il tempo in cui è stato cambiato lo stato del `TODO`.

Org mode può anche registrare automaticamente un timestamp e opzionalmente una nota quando segni un elemento `TODO` come `DONE`, o ogni volta che cambi lo stato di un elemento `TODO`. Questo sistema è altamente configurabile, le impostazioni possono essere basate su una parola chiave e possono essere localizzate in un file o ogni sotto-albero. Per maggiori informazioni su come cronometrare il tempo di lavoro vedi 9.4.

Chiudere un elemento

La più semplice registrazione è tenere traccia di *quando* un certo elemento TODO è segnato come completato. Questo può essere fatto con⁴

```
(setq org-log-done 'time)
```

Nel momento in cui modifichi lo stato di un oggetto da TODO (non fatto) in uno degli stati DONE, una linea **CLOSED: [timestamp]** viene inserita subito sotto l'intestazione.

Se vuoi registrare una nota con il timestamp usa⁵

```
(setq org-log-done 'note)
```

Ti verrà mostrato un prompt dove inserire la nota e questa nota sarà registrata sotto l'oggetto con un'intestazione **Closing Note**.

Tenere traccia degli stati TODO

Potresti voler tenere traccia dello stato del TODO. Puoi registrare solo un timestamp o un timestamp con nota per un cambiamento. Queste registrazioni sono inserite dopo il titolo come elementi di lista. Quando prendi una gran quantità di note, potresti volere tenere le note esternamente in un drawer. Per questa funzionalità customizza la variabile **org-log-into-drawer**.

Per la registrazioni dello stato, Org mode si basa sulla configurazione delle parole chiave. Questo si ottiene aggiungendo un marker speciale ! (per il timestamp) e @ (per le note) tra parentesi dopo ogni parola chiave. Per esempio:

```
#+TODO: TODO(t) WAIT(w@/!) | DONE(d!) CANCELLED(c@)
```

definendo le keyword TODO e le chiavi per l'accesso rapido, è anche richiesto che un tempo sia registrato quando un oggetto viene settato a DONE, e che una nota sia registrata quando si cambia a WAIT o CANCELED. La stessa sintassi funziona anche quando setti **org-todo-keywords**.

⁴L'impostazione corrispondente nel buffer è **#+STARTUP: logdone**.

⁵L'impostazione corrispondente nel buffer è **#+STARTUP: lognotedone**.

6.4 Priorità

Se usi la modalità Org in maniera estesa, potresti trovarti con tanti elementi TODO e quindi potrebbe aver senso dargli delle priorità. Si possono dare priorità mettendo un *priority cookie* nell'intestazione dell'elemento TODO, tipo questo:

```
*** TODO [#A] Write letter to Sam Fortune
```

Org mode supporta tre priorità A, B e C. A è la più alta, B quella di default se non specificata. Le priorità creano una differenza solo nell'agenda.

C-c , (**org-priority**) Imposta la priorità nell'intestazione corrente. Premi A, B o C per selezionare la priorità o SPC per rimuovere il cookie.

S-SU (**org-priority-up**); **S-GIU** (**org-priority-down**) Aumentano/diminuiscono la priorità dell'intestazione corrente.

6.5 Dividi i compiti in sotto compiti

A volte é consigliabile dividere i grandi compiti in compiti più piccoli, e gestibili. Puoi farlo creando una struttura ad albero con sotto-elementi TODO, con dettagliati i sotto-compiti dell'albero. Per avere un'anteprima della frazione dei sotto-compiti che hai già segnato come fatti, inserisci [/] o [%] dove vuoi nell'intestazione. Questi cookie saranno aggiornati ogni volta che lo stato dei TODO figli cambia o quando premi C-c C-c nel cookie. Per esempio:

```
* Organizzare un party [33%]  
** TODO Chiamare le persone [1/2]  
*** TODO Peter  
*** DONE Sarah  
** TODO Buy food  
** DONE Parlare con il vicino
```

6.6 Liste selezionabili

Ogni elemento in una lista (vedi 3.6) può essere trasformata in una casella di controllo (checkbox) che deve iniziare con la stringa []. Le checkbox non sono incluse nella lista di TODO globale, sono molto comode per dividere un compito in un numero di semplici step.

Qui c'è un esempio di una lista di checkbox.

- * TODO Organizzare un party [2/4]
 - [-] Chiamare le persone
 - [] Peter
 - [X] Sarah
 - [X] Ordinare il cibo

Le caselle di controllo lavorano in maniera gerarchica, quindi se un elemento checkbox ha figli che sono checkbox, attivando una casella di controllo figlia si fa in modo che nella casella di controllo padre venga riflesso se nessuno, alcuni o tutti i figli sono stati selezionati.

I seguenti comandi operano con le checkbox:

C-c C-c, C-u C-c C-c (`org-toggle-checkbox`) Aggiorna lo stato di una checkbox o—con un argomento prefisso—cambia la presenza della checkbox nel punto.

M-S-RET (`org-insert-todo-heading`) Inserisce un nuovo elemento con una checkbox. Questo funziona solo se ci si trova in un elemento della lista (vedi 3.6).

7 TAG

Un modo eccellente per implementare le label ed i contesti, per le informazioni incrociate correlate, é quello di assegnare dei *tag* alle intestazioni. La modalità Org ha un supporto molto esteso per i tag.

Ogni intestazione può contenere una lista di tag; sono messi alla fine delle intestazioni. I tag sono normali parole che contengono lettere, numeri, _ e @. I tag sono preceduti e succeduti da un singolo due-punti, es. `:work:`. Alcuni tag possono essere specificati come in `:work:urgent:`. I tag di default sono in grassetto dello stesso colore dell'intestazione.

Ereditare i tag

I tag usano la struttura gerarchica dell'albero di outline. Se un'intestazione ha un certo tag, tutti i sottotitoli ereditano il tag. Per esempio nella lista

```
* Incontro con il gruppo Francese      :work:
** Sommario di Frank                   :boss:notes:
*** TODO Prepara le slide per lui      :action:
```

l'intestazione finale ha i tag `work`, `boss`, `notes` ed `action` anche se nel titolo finale non vengono segnati questi tag.

Puoi anche impostare i tag che tutte gli elementi nel file devono ereditare semplicemente come se questi tag fossero definiti in un ipotetico livello zero che sovrasta l'intero file. Usando una linea del genere⁶:

```
#+FILETAGS: :Peter:Boss:Secret:
```

Impostare i tag

I tag possono essere semplicemente inserite nel buffer alla fine di una intestazione. Dopo i due-punti, M-TAB offre il completamento dei tag. Ci sono anche altri comandi speciali per inserire i tag:

C-c C-q (`org-set-tags-command`) Inserisce un nuovo tag nel titolo corrente. Org mode, inoltre, offre il completamento o una speciale singola-chiave di interfaccia per impostare i tag, vedi sotto.

C-c C-c (`org-set-tags-command`) Quando il punto è in un'intestazione, questo compie la stessa azione di C-c C-q.

Org supporta l'inserimento dei tag da una *lista di tag*. Di default questa lista è costruita dinamicamente, contiene tutti i tag usati, al momento, nel buffer. Puoi anche specificare una lista fissa globale con la variabile `org-tag-alist`. In fine puoi impostare i tag di default da mettere in un file usando la keyword `TAGS`, tipo

```
#+TAGS: @work @home @tennisclub
#+TAGS: laptop car pc sailboat
```

Di default Org mode usa la funzionalità standard di completamento del mini-buffer per inserire i tag. Comunque, implementa anche un'altro metodo di selezione, veloce, del tag chiamato *fast tag selection*. Questo ti permette di selezionare e deselectare i tag con solo una pressione di una singola chiave. Per questo scopo è bene assegnare una singola lettera ai tuoi tag più comunemente usati. Puoi fare questo globalmente configurando la variabile `org-tag-alist` nel tuo file di init di Emacs. Per esempio, puoi trovare utile assegnare a degli elementi in diversi file con `@home`. In questo caso puoi impostare qualcosa di simile:

⁶Come tutto nelle impostazioni del buffer, premendo C-c C-c si attivano i cambiamenti nella linea.

```
(setq org-tag-alist '(("@work" . ?w) ("@home" . ?h) ("laptop" . ?l)))
```

Se il tag é rilevante solo per il file su cui stai lavorando, allora puoi inserire ed impostare la keyword **TAGS** come:

```
#+TAGS: @work(w) @home(h) @tennisclub(t) laptop(l) pc(p)
```

Raggruppare i tag

Un tag può essere definito come un *gruppo di tag* per un insieme di altri tag. Il gruppo di tag può essere visto come il “termine più ampio” per un insieme di tag.

Puoi impostare un gruppo di tag usando le parentesi ed inserendo i due-punti tra il gruppo di tag ed i tag relazionati:

```
#+TAGS: [ GTD : Control Presp ]
```

o se i tag nel gruppo sono mutualmente esclusivi:

```
#+TAGS: { Context : @Home @Work }
```

Quando cerchi per un gruppo di tag, ti vengono ritornati tutti i membri del gruppo e i sottogruppi. Nella visualizzazione agenda, filtrando per un gruppo di tag mostra o nasconde i titoli segnati con almeno uno dei membri, del gruppo o ognuno dei sottogruppi.

Se vuoi, temporaneamente, ignorare i gruppi di tag, cambia il gruppo dei tag supportati con **org-toggle-tags-groups**, mappato con C-c C-x q.

Ricerca dei tag

C-c / m o C-c \ (~ org-match-sparse-tree~) Crea un albero sparso con tutte le intestazioni che corrispondono ad un tag cercato. Con un argomento prefisso C-u, ignora le intestazioni che non sono una linea TODO.

M-x org-agenda m (org-tags-views) Crea una lista globale di tag corrispondenti, da tutti i file dell’agenda. vedi 11.5.

M-x org-agenda M (org-tags-views) Crea una lista globale di tag corrispondenti, da tutti i file dell’agenda, ma controlla solo gli elementi TODO.

Tutti questi comandi richiedono una stringa di corrispondenza che rispetti la logica booleana di base `+boss+urgent-project1`, per trovare gli elementi con i tag `boss` e `urgent`, ma non `project`, oppure `Kathy|Sally` per trovare un elemento che sia segnato come `Kathy` o `Sally`. La sintassi completa per cercare queste stringhe è ricca e permette anche la ricerca all'interno di keyword `TODO`, di livelli e di proprietà. Per una più dettagliata descrizione con diversi esempi, vedi 11.5.

8 Proprietà

Le proprietà sono un'associazione di chiave-valore associate ad un oggetto. Risiedono in speciali drawer con nome `PROPERTIES`. Ogni proprietà è specificata su una linea singola, con la chiave prima (circondata dai due punti) ed il valore dopo.

```
* CD collection
** Classic
*** Goldberg Variations
    :PROPERTIES:
    :Title:      Goldberg Variations
    :Composer:   J.S. Bach
    :Publisher:  Deutsche Grammophon
    :NDisks:     1
    :END:
```

Potresti definire i valori permessi per una particolare proprietà `XYZ` settando una proprietà `XYZ_ALL`. Questa speciale proprietà è *ereditata*, quindi se lo imposti nell'oggetto di livello 1, viene applicato agli oggetti dell'albero. Quando i valori permessi sono definiti, l'impostazione della corrispondente proprietà diventa più semplice e meno soggetta ad errori di battitura. Per l'esempio con la collezione di CD, possiamo predefinire la casa discografica ed il numero di dischi nella custodia tipo:

```
* CD collection
    :PROPERTIES:
    :NDisks_ALL:  1 2 3 4
    :Publisher_ALL: "Deutsche Grammophon" Philips EMI
    :END:
```

Se vuoi impostare una proprietà che può essere ereditata da ogni oggetto nel file, usa una linea tipo:

`#+PROPERTY: NDisks_ALL 1 2 3 4`

Il seguente comando aiuta a lavorare con le proprietà:

C-c C-x p (`org-set-property`) Imposta una proprietà. Questo richiede un nome ed un valore.

C-c C-c d (`org-delete-property`) Rimuove una proprietà dall'oggetto corrente.

Per creare un albero sparso e liste speciali con la selezione basata sulle proprietà, gli stessi comandi sono usati per cercare tag (vedi 7). La sintassi per la stringa di ricerca è descritta nella sezione 11.5.

9 Data e orario

Per assistere la pianificazione, gli elementi TODO possono essere etichettati con una data e/o un orario. La speciale stringa formattata trasporta le informazioni di data ed orario chiamata, in Org mode, timestamp. Questo può creare un po di confusione in quanto il timestamp é usato, a volte, per indicare quando qualcosa é stato creato o l'ultima modifica. Comunque, nella modalità Org questo termine viene usato in questo senso più ampio.

Il timestamp può essere usato per pianificare appuntamenti, schedulare impegni, impostare scadenze, tracciare il tempo ed altro. La sezione seguente descrive il formato timestamp e gli strumenti che Org mode ci mette a disposizione per i casi d'uso che riguardano il tempo e gli intervalli di tempo.

9.1 Timestamp

Un timestamp é una specifica data—possibilmente con un orario o un range di orari—in uno speciale formato, come `<2003-09-16 Tue>` o `<2003-09-16 Tue 09:39>` oppure `<2003-09-16 Tue 12:00-12:30>`. Un timestamp può essere inserito ovunque nelle intestazioni o nel corpo di un elemento dell'albero Org. La sua presenza genera la visualizzazione di un elemento in una specifica data nell'agenda (vedi 11.3). Distinguiamoli:

Timestamp semplici; Eventi; Appuntamenti Un semplice timestamp che assegna ad un elemento una data/orario. Questo é come scrivere un appuntamento o un evento in un'agenda di carta.

```
* Incontro con Peter al cinema
  <2006-11-01 Wd 19:15>
* Discussione sui cambiamenti climatici
  <2006-11-02 Thu 20:00-22:00>
* Giorni di ferie
  <2006-11-03 Fri>
  <2006-11-06 Mon>
```

Timestamp con un intervallo che si ripete Un timestamp contenente un *intervallo di ripetizione*, indica che, lo stesso, non si ripeta solo una volta ma ancora e ancora dopo un certo intervallo di N ore (h), giorni (d), settimane (w), mesi (m) o anni (y). Di seguito ne mostriamo uno che si ripete ogni Mercoledì:

```
* Prendere Sam da scuola
  <2007-05-16 Wed 12:30 +1w>
```

Espressioni per gli elementi in stile diario Per specificare diverse date complesse, Org mode supporta l'uso di speciali espressioni di elementi diario, implementati nel pacchetto Emacs Calendar. Per esempio, con un tempo opzionale

```
* 22:00-23:00 L'incontro dei nerd ogni secondo Giovedì del mese
  <%%(diary-float t 4 2)>
```

Range di tempo Un range é un timestamp connesso da due -.

```
* Discussione sui cambiamenti climatici
  <2006-11-02 Thu 10:00-12:00>
```

Range di Tempo/Data Due timestamp connessi da -- denotano un range. Nell'agenda, l'intestazione é mostrata nella prima e nell'ultima data del range, e in ogni data che rientra nel range. Il primo esempio é specificato soltanto le data del range invece nel secondo esempio viene specificato un intervallo di tempo per entrambe le date.

```
** Incontro ad Amsterdam
  <2004-08-23 Mon>--<2004-08-26 Thu>
** Riunioni del comitato di questa settimana
  <2004-08-23 Mon 10:00-11:00>--<2004-08-26 Thu 10:00-11:00>
```

Timestamp inattivo Come un semplice timestamp, ma con le parentesi quadre al posto delle angolari. Questo timestamp sarà inattivo nel senso che *non* innescherà un elemento da mostrare nell’agenda.

```
* Gillian arriva tardi per la quinta volta  
[2006-11-01 Wed]
```

9.2 Creare un timestamp

Per fare in modo che Org mode possa riconoscere i timestamp devono essere in uno specifico formato. Tutti i comandi descritti sotto producono un timestamp nella forma corretta.

C-c . (org-timestamp) Richiede una data ed inserisce un corrispondente timestamp. Quando il punto è in un timestamp preesistente nel buffer, il comando viene usato per modificare questo timestamp invece di inserirne uno nuovo. Quando questo comando viene usato due volte in successione un range di tempo viene inserito. Con un argomento prefisso aggiunge anche l’orario corrente.

C-c ! (org-timestamp-inactive) Come C-c ., ma inserisce un timestamp inattivo che non genera un elemento agenda.

S-SINISTRA (org-timestamp-down-day), S-DESTRA (org-timestamp-up-day)
Cambia la data, di un giorno, nel punto corrente.

S-SU (org-timestamp-up), S-GIU (org-timestamp-down) All’inizio o sulle parentesi graffe di un timestamp, ne cambia il tipo. All’interno di un timestamp ne modifica l’elemento al punto. Il punto può essere un anno, un mese, un giorno, un’ora o un minuto. Quando il timestamp contiene un intervallo di tempo tipo 15:30-16:30, modificando il primo verrà modificato anche il secondo, spostando, il blocco di tempo, di lunghezza costante. Per cambiare la lunghezza si deve modificare il secondo orario.

Quando Org mode richiede una data/orario, accetta qualsiasi stringa contenente delle informazioni su data e/o orario e, intelligentemente, interpreta la stringa, ricavando valori predefiniti per le informazioni non specificate dalla data ed ora correnti. Puoi anche selezionare una data nel calendario pop-up. Consulta il manuale per ulteriori informazioni su come funziona esattamente la richiesta di data/orario.

9.3 Scadenze e schedulazioni

Un timestamp potrebbe essere preceduto da delle speciali keyword per facilitarne la pianificazione:

C-c C-d (org-deadline) Inserisce la parola chiave `DEADLINE` assieme al timestamp, nella linea dopo l'intestazione.

Significato: l'attività—molto probabilmente una voce `TODO`, anche se non necessariamente—si suppone sia stata completata in quella data.

Nella data di scadenza, il task é mostrato nell'agenda. Inoltre, l'agenda per *oggi* mostra un avviso riguardante l'approssimarsi o la mancata scadenza, a partire da `org-deadline-warning-days` prima della data di scadenza e continuando fino a che l'elemento non sarà segnato come fatto. Un esempio:

```
*** TODO Scrivere un articolo sulla Terra per la Guida
      DEADLINE: <2004-02-29 Sun>
      L'editore in carica e' [[bbdb:Ford Prefect]]
```

C-c C-s (org-schedule) Inserisce la parola `SCHEDULED` con il timestamp sotto la linea dell'intestazione.

Significato: Hai pianificato di iniziare quel lavoro in quella data⁷

L'intestazione é mostrata nella data⁸ impostata. In aggiunta, nella compilazione per *oggi*, é presente un promemoria che la data pianificata é trascorsa, fino a che l'elemento non sarà segnato come fatto, ovvero l'attività viene automaticamente inoltrata fino al completamento.

```
*** TODO Chiamare Trillian per un appuntamento a New Years Eve.
      SCHEDULED: <2004-12-25 Sat>
```

Spesso i compiti hanno la necessità di essere ripetuti ancora ed ancora. Org mode aiuta nell'organizzazione di questi lavori usando un, così detto, repeater in una `DEADLINE`, `SCHEDULED` o in un timestamp semplice. Nel seguente esempio:

⁷Questo é completamente diverso da ciò che normalmente si intende *pianificare un appuntamento*, che in Org viene eseguita semplicemente inserendo un timestamp senza keyword.

⁸Sarà comunque mostrato in quella data anche se verrà contrassegnato come fatto. Se non ti piace, imposta la variabile `org-agenda-skip-scheduled-if-done`.

**** TODO Paga l'affitto**
DEADLINE: <2005-10-01 Sat +1m>

il +1m é un repeater; quello che si intende é che il lavoro ha una data di scadenza nel <2005-10-01> e si ripete ogni (un) mese partendo da quella data.

9.4 Cronometra il tempo di lavoro

La modalità Org ti permette di cronometrare quanto tempo ti serve per uno specifico progetto.

C-c C-x C-i (org-clock-in) Fa partire il cronometro nell'elemento corrente (clock-in). Questo inserisce la parola chiave **CLOCK** assieme ad un timestamp. Quando viene richiamato assieme ad un argomento prefisso, seleziona il task da una lista di task chiusi di recente.

C-c C-x C-o (org-clock-out) Ferma il timer (clock-out). Questo inserisce un'ulteriore timestamp nella stessa posizione dove il cronometro aveva inserito l'ultimo inizio. Viene anche direttamente calcolato ed inserito il tempo dopo il range di tempo nella forma =>HH:MM.

C-c C-x C-e (org-clock-modify-effort-estimate) Aggiorna lo sforzo richiesto per il corrente timer di lavoro.

C-c C-x C-q (org-clock-cancel) Cancella il cronometro corrente. Questo é utile se il cronometro é partito per sbaglio o se finisci di lavorare o altro.

C-c C-x C-j (org-clock-goto) Salta all'intestazione del lavoro, nel corrente timer in lavoro. Con un argomento prefisso C-u seleziona il lavoro targhet da una lista di lavori cronometrati di recente.

La chiave l viene solitamente usata nell'agenda (vedi 11.3), mostra su quali task stai lavorando o quelli che hai chiuso durante il giorno.

10 Cattura, sposta e archivia

Una parte importante di ogni sistema di organizzazione é l'abilità di catturare rapidamente nuove idee e task, e associare il materiale di riferimento con esse. Org lo fa usando un processo detto *capture*. Può anche archiviare i relativi file di un lavoro (*allegati*) in una speciale directory. Una volta nel sistema,

le attività ed i processi devono essere spostati. Muovere completamente un albero di progetto in un archivio di file rende il sistema compatto e veloce.

10.1 Cattura

Capture ti permette di archiviare velocemente le note con una piccola interruzione del tuo processo di lavoro. Puoi definire modelli per i nuovi elementi ed associarli con differenti target per salvare le note.

Impostare capture

L'impostazione seguente setta un file target⁹ di default per le note.

```
(setq org-default-notes-file (concat org-directory "/notes.org"))
```

Puoi anche definire una combinazione globale per catturare nuovo materiale (vedi Attivazione).

Usare capture

m-x org-capture (org-capture) Inizia il processo di cattura, ti posiziona in un buffer limitato per la modifica.

C-c C-c (org-capture-finalize) Uno volta che hai finito di inserire le informazioni nel buffer di cattura, C-c C-c ti riporta nella finestra da dove hai fatto partire il processo, in modo che tu possa riprendere il lavoro senza altre distrazioni.

C-c C-w (org-capture-refile) Finalizza il processo di cattura mettendo le note in una differente posizione (vedi 10.2).

C-c C-k (org-capture-kill) Annulla il processo di cattura e ti riporta allo stato precedente.

Modelli di cattura

Puoi usare i modelli per differenti tipi di oggetti di cattura, e per differenti posizioni. Per dire se vuoi usare un template per creare degli oggetti TODO generali, e se vuoi mettere questi elementi sotto un intestazione **Tasks** nel tuo file `~/org/gtd.org`. Puoi anche catturare degli oggetti giornalieri come un

⁹Usando i modelli di capture, puoi avere un controllo più preciso sulla posizione della cattura. vedi Modelli di cattura

albero di date nel file `journal.org`. Una possibile configurazione potrebbe essere simile a questa:

```
(setq org-capture-templates
  '(("t" "Todo" entry (file+headline "~/org/gtd.org" "Tasks")
    "* TODO %?\n %i\n %a")
    ("j" "Journal" entry (file+datetree "~/org/journal.org")
    "* %?\nEntered on %U\n %i\n %a"))))
```

Se premi `t` quando sei nel menu di capture, Org preparerà il template per te in questo modo:

```
* TODO
[[file:COLLEGAMENTO DI DOVE HAI INIZIATO LA CATTURA]]
```

Durante l'espansione del template, il carattere speciale di escape `%`¹⁰ permette l'inserimento dinamico del contenuto. Qui c'è una piccola selezione di possibilità, consulta il manuale per averne di più.

<code>%a</code>	annotazione, di norma il link creato con <code>org-store-link</code>
<code>%i</code>	contenuto iniziale, la regione dove capture è chiamata con <code>C-u</code>
<code>%t %T%</code>	timestamp, solo data o data ed orario.
<code>%u %U</code>	come sopra ma per i timestamp inattivi
<code>%?</code>	dopo avere completato il template, posiziona il punto li.

10.2 Sposta e Copia

Quando revisioni i dati catturati, potresti volere spostare o copiare alcuni dei tuoi elementi in una lista differente, per esempio in un progetto. Tagliare, cercare la posizione corretta e poi incollare la nota è scomodo. Per semplificare il processo puoi usare i comandi speciali che trovi di seguito:

C-c C-w (org-agenda-refile) Sposta gli oggetti o le regioni in un punto. Questo comando offre possibili posizioni dove spostare gli oggetti e ti permette di selezionarne uno con il completamento. L'oggetto (o tutti gli oggetti nella regione) è inserito sotto l'intestazione target come sotto oggetti.

Di default, tutti i livelli 1 delle intestazioni nel buffer corrente sono considerati come target, ma puoi avere delle definizioni molto più complesse riguardo un certo numero di file. Vedi la variabile `org-refile-targets` per dettagli.

¹⁰Se hai bisogno di una di queste sequenze letterali, usa il carattere di escape `\` con `%`.

C-c C-c C-w (org-agenda-refile) Usa l'interfaccia di spostamento per saltare ad una intestazione.

C-c Cu C-c C-w (org-refile-goto-last-stored) Salta ad una posizione dove **org-refile** ha spostato l'albero l'ultima volta.

C-c M-w (org-refile-copy) La copia funziona come lo spostamento, fatta eccezione che le note originali non vengono cancellate.

10.3 Archiviare

Quando un progetto rappresentato da un (sotto)albero é concluso, potresti volerlo spostare fuori dai piedi e interrompere i suoi contributi all'agenda. L'archiviazione é importante per tenere i tuoi file di lavoro compatti e la vista della ricerca globale dell'agenda più veloce.

La più comune azione di archiviazione é di spostare un albero di progetto in un altro file, il file di archivio.

C-c C-x C-a (org-archive-subtree-default) Archivia il corrente elemento usando un comando specificato nella variabile **org-archive-default-command**.

C-c C-x C-s o brevemente C-c \$ (org-archive-subtree) Archivia il sotto-albero partendo dalla posizione del punto nella posizione data da **org-archive-location**.

La posizione di archiviazione standard é un file nella stessa directory del file corrente, con il nome derivato aggiungendo **_archive** al nome del file corrente. Puoi anche scegliere sotto quale intestazione archiviare gli elementi, con la possibilità di aggiungerli ad un albero delle date in un file. Per informazioni ed esempi su come specificare il file e le intestazioni, vedi la documentazione sulla stringa della variabile **org-archive-location**.

C'è anche un'opzione nel buffer per settare questa variabile, per esempio:

```
#+ARCHIVE: %s_done::
```

11 Visualizzazioni Agenda

A causa del modo in cui lavora Org, gli elementi TODO, gli elementi di timestamp ed i titoli taggati possono essere presenti nel file o in file diversi. Per avere una visione delle azioni in corso o degli eventi che sono importanti per una data particolare, queste informazioni devono essere raccolte, ordinate e visualizzate in modo organizzato.

Le informazioni estratte sono mostrate in uno speciale *buffer agenda*. Questo buffer é in sola lettura, ma fornisce comandi per visitare le corrispondenti posizioni nei file Org originali e anche modificare questi file da remoto. La modifica remota dal buffer agenda significa, per esempio, che puoi cambiare le date di una scadenza e gli appuntamenti da quel buffer agenda. Per una lista di comandi avviabili nel buffer Agenda, vedi 11.7.

11.1 File agenda

L'informazione mostrata è normalmente raccolta da tutti i *file agenda*, i file sono listati nella variabile `org-agenda-files`.

C-c [(org-agenda-file-to-front) Aggiunge il file corrente nella lista dei file agenda. Il file è aggiunto all'inizio della lista dei file. Se si trova già nella lista, viene spostato all'inizio. Con un argomento prefisso, il file viene aggiunto/spostato alla fine.

C-c] (org-remove-file) Rimuove il corrente file dalla lista dei file agenda.

C-' / C-\ (org-cycle-agenda-files) Cicla tra la lista dei file agenda, visita un file dopo l'altro.

11.2 Il Distributore Agenda

Le viste sono create attraverso un distributore, accessibile con M-x `org-agenda`, o, meglio se abbinato ad una chiave globale (vedi Attivazione). Mostra un menu in cui una lettera aggiuntiva viene richiesta per eseguire il comando. Il distributore offre i seguenti comandi predefiniti:

a Crea un agenda tipo calendario (vedi 11.3).

t, T Crea una lista di tutti gli elementi TODO (vedi 11.4).

m, M Crea una lista di intestazioni che corrispondono all'espressione data (vedi 11.5).

s Crea una lista di oggetti selezionati tramite un'espressione booleana di keyword e/o espressioni regolari che devono o non devono verificarsi negli oggetti.

11.3 L'agenda Settimanale/Giornaliera

Il proposito dell'*agenda* settimanale/giornaliera é di funzionare come una pagina di un agenda cartacea, mostrando tutti i lavori per la settimana od il giorno corrente.

M-x org-agenda a (org-agenda-list) Compila una agenda per la settimana corrente da una lista di file Org. L'agenda mostra gli oggetti per ogni giorno.

Org mode capisce la sintassi del diario e ti permette di usare degli elementi espressione agenda direttamente nei file Org:

```
* Holidays
:PROPERTIES:
:CATEGORY: Holiday
:END:
%(org-calendar-holiday)    ; Funzione speciale per il nome holiday

* Birthdays
:PROPERTIES:
:CATEGORY: Ann
:END:
%(org-anniversary 1956  5 14) Arthur Dent is %d years old
%(org-anniversary 1869 10  2) Mahatma Gandhi would be %d years old
```

Org può interagire con la funzione di modifica degli appuntamenti di Emacs. Per aggiungere un appuntamento ai tuoi file agenda, usa il comando `org-agenda-to-appt`.

11.4 Lista globale TODO

La lista globale dei TODO contiene tutti gli oggetti TODO, non completati, formattati e raggruppati in un singolo punto. La modifica remota degli elementi TODO ti dà la possibilità di cambiare lo stato di un oggetto TODO con una singola pressione di un tasto. Per i comandi avviabili nella TODO list, vedi 11.7.

M-x org-agenda t (org-todo-list) Mostra la lista globale dei TODO. Questo raggruppa gli elementi TODO da tutti i file agenda (vedi 11) in un unico buffer.

M-x org-agenda T (org-todo-list) Come sopra, ma permette la selezione di uno specifica keyword TODO.

11.5 Trova i tag e le proprietà

Se le intestazioni nel file agenda sono marcati con i *tags* (vedi 7), o hanno delle proprietà (vedi 8), puoi selezionare le intestazioni in base a questi meta-dati e raccoglierli in un buffer agenda. La stessa cosa qui descritta é applicabile anche quando crei gli alberi sparsi con C-c / m.

M-x org-agenda m (org-tags-views) Produce una lista di tutte le intestazioni che trova con un dato set di tag. Il comando richiede l'inserimento di un criterio di selezione, che é un espressione logica booleana con tag, come `+work+urgent-witboss` o `work|home` (vedi 7). Se usi spesso una specifica ricerca, puoi definire un comando personalizzato (vedi 11.2).

M-x org-agenda M(org-tags-view) Come m ma seleziona solo le intestazioni che sono oggetti TODO.

Una stringa di ricerca può usare operatori Booleani `&` per AND e `|` per OR. `&` si lega più fortemente di `|`. Le parentesi non sono ancora implementate. Ogni elemento nella ricerca ha o un tag, un'espressione regolare che trova un tag o un'espressione tipo `PROPERTY OPERATOR VALUE` con un operatore di comparazione, accede ad una valore proprietà. Ogni elemento dev'essere preceduto da `-` per non selezionarlo e un `+` che é zucchero sintattico per la selezione positiva. L'operatore AND `&` é opzionale quando `+` o `-` sono presenti. Ecco alcuni esempi, usando solo i tag.

`+work-boss` Seleziona le intestazioni taggate con `work`, ma scarta quelle che sono marcate `boss`

`work|laptop` Seleziona le linee taggate `work` o `laptop`.

`work|laptop+night` Come sopra ma richiede che le linee `laptop` siano anche segnate come `night`.

Puoi anche cercare le proprietà e i tag nello stesso tempo, vedi il manuale per informazioni aggiuntive.

11.6 Vista di ricerca

Questa vista agenda é una funzione di ricerca generale del testo per gli oggetti Org mode. É particolarmente utile per trovare le note.

M-x org-agenda s (org-search-view) Questo é una speciale ricerca che permette di selezionare gli elementi che corrispondono ad una sottostringa o a specifiche voci usando la logica booleana.

Per esempio, la stringa di ricerca `computer equipment` trova gli elementi che contengono `computer equipment` come una sotto stringa.

La vista di ricerca può anche ricercare per specifiche keyword negli oggetti, usando la logica booleana. La stringa di ricerca `+computer +wifi -ethernet -{8'.11[bg]}` trova le note che contengono le parole chiave `computer` e `wifi` ma non la keyword `ethernet` e che non corrispondono all'espressione regolare `8\'.11[bg]`. che esclude sia `8.11b` che `8.11g`.

Nota che in aggiunta ai file agenda, questo comando cerca anche nei file elencati in `org-agenda-text-search-extra-files`.

11.7 Comandi nel Buffer Agenda

Gli oggetti nel buffer agenda sono un rimando ai file Org o ai file del diario da dove vengono generati. Non ti é permesso modificare lo stesso buffer agenda, ma i comandi ti permettono di vedere e di saltare agli oggetti nella posizione originale e modificare i file Org originarli da “remoto” partendo dal buffer agenda. Questa é solo una selezione dei principali comandi, esplora il menu agenda ed il manuale per la lista completa.

Movimento

n (`org-agenda-next-line`) Linea successiva (come GIU e C-n).

p (`org-agenda-previous-line`) Linea precedente (come SU e C-p).

Vedere/Andare ad un file Org

SPC (`org-agenda-show-and-scroll-up`) Mostra la posizione originale dell'elemento in un'altra finestra. Con un argomento prefisso, si assicura che i drawers rimangano chiusi.

TAB (`org-agenda-goto`) Va alla posizione originale dell'oggetto in un'altra finestra.

RET (`org-agenda-switch-to`) Va alla posizione originale dell'oggetto e chiude le altre finestre.

Cambia visualizzazione

- o** (`delete-other-windows`) Elimina le altre finestre.
- v d o brevemente d** (`org-agenda-day-view`) Passa alla vista del giorno
- v w o brevemente w** (`org-agenda-week-view`) Passa alla vista della settimana
- f** (`org-agenda-later`) Va avanti nel tempo per visualizzare l'intervallo successivo a quello corrente. Per esempio, se la visualizzazione copre la settimana, passa alla settimana seguente.
- b** (`org-agenda-earlier`) Torna indietro nel tempo per visualizzare le date precedenti.
- .** (`org-agenda-goto-today`) Va ad oggi.
- j** (`org-agenda-goto-date`) Richiede una data e ci va.
- v l or v L o brevemente l** (`org-agenda-log-mode`) Attiva/disattiva la modalità Logbook. Nella modalità Logbook gli oggetti che sono segnati come completati, mentre era attiva la registrazione (vedi la variabile `org-log-done`), sono mostrati nell'agenda, così come le voci che sono state cronometrate in quel giorno. Quando chiamato con un argomento prefisso C-u, mostra tutti i possibili oggetti logbook, includendo i cambi di stato.
- r, g** (`org-agenda-redo`) Ricarica il buffer agenda, per esempio per aggiornare i cambiamenti dopo una modifica del timestamp degli items.
- s** (`org-save-all-org-buffers`) Salva tutti i buffer Org nella sessione di Emacs corrente e anche la posizione degli ID.

Modifica remota

0-9 Argomento numerico

- t** (`org-agenda-todo`) Cambia lo stato TODO dell'elemento, sia nell'agenda che nel file di origine.
- C-k** (`org-agenda-kill`) Cancella l'oggetto corrente dall'agenda e lungo l'intero sotto-albero nel file Org originale.
- C-c C-w** (`org-agenda-refile`) Muove l'oggetto al punto.

a (`org-agenda-archive-default-with-confirmation`) Archivia il sotto-albero corrispondente all'elemento del punto usando il comando di archiviazione di default impostato con nella variabile `org-archive-default-command`.

\$ (`org-adenda-archive`) Archivia il sotto-albero corrispondente all'intestazione corrente.

C-c C-s (`org-agenda-schedule`) Schedula questo elemento. Con un argomento prefisso, rimuove il timestamp schedulato.

C-c C-d (`org-agenda-deadline`) Imposta una scadenza per questo elemento. Con un argomento prefisso, rimuove la scadenza.

S-DESTRA (`org-agenda-do-date-later`) Cambia il timestamp associato alla linea corrente con un giorno nel futuro.

S-SINISTRA (`org-agenda-do-date-earlier`) Cambia il timestamp associato alla linea corrente con un giorno nel passato.

I (`org-agenda-clock-in`) Inizia il cronometro nell'elemento corrente.

O (`org-agenda-clock-out`) Ferma il cronometro precedentemente attivato.

X (`org-agenda-clock-cancel`) Cancella il cronometro attivo.

J (`org-agenda-clock-goto`) Va all'orologio corrente in un'altra finestra.

Chiudi ed esci

q (`org-agenda-quit`) Chiude l'agenda, rimuove il buffer agenda

x (`org-agenda-ecit`) Esce dall'agenda, rimuove il buffer agenda e tutti i buffer caricati da Emacs per la compilazione dell'agenda.

11.8 Personalizzare la vista agenda

La prima applicazione di personalizzazione di ricerca é la definizione di una combinazione per la ricerca usata più frequentemente, o creando un ordine del giorno oppure uno sparse tree (la lettera copre soltanto il buffer corrente).

I comandi personalizzati sono configurati nella variabile `org-agenda-custom-commands`. Puoi personalizzare questa variabile, per esempio premendo C dal dispatcher agenda (vedi 11.2). Puoi anche impostarla direttamente con Emacs Lisp nel file di init di Emacs. Il seguente esempio contiene tutte viste valide dell'agenda.

```
(setq org-agenda-custom-commands
      '(("w" todo "WAITING")
        ("u" tags "+boss-urgent")
        ("v" tags-todo "+boss-urgent")))
```

La stringa iniziale in ogni elemento definisce le chiavi che puoi premere dopo l'avvio del comando del dispatcher per poter accedere al comando. Solitamente questo é di un solo carattere. Il secondo parametro é il tipo di ricerca, seguito dalla stringa o espressione regolare da usare per la selezione. L'esempio sopra quindi definirà:

- w** Come per la ricerca globale per gli oggetti TODO con **WAITING** come parola chiave del TODO
- u** Come per la ricerca globale di tags per le intestazioni segnate **boss** ma non **urgent**.
- v** La stessa ricerca, ma limitata per le intestazioni che sono anche elementi TODO.

12 Markup per contenuti ricchi

Org soprattutto riguarda l'organizza e la ricerca all'interno delle tue note di testo semplice. Comunque, fornisce un leggero ma robusto linguaggio di markup per la formattazione del testo ed altro. Usato assieme al framework di esportazione (vedi 13), con Org puoi produrre documenti molto belli.

12.1 Paragrafi

I paragrafi sono separati tra loro da una linea vuota. Se hai bisogno di forzare un'interruzione di linea all'interno di un paragrafo, usa `\\` alla fine della linea.

Per preservare l'interruzione di linea, l'indentazione e le linee bianche in una regione, ma anche usare la formattazione normale, puoi usare questo costruito, puoi anche usarlo per formattare la poesia.

```
#+BEGIN_VERSE
Great clouds overhead
Tiny black birds rise and fall
Snow covers Emacs
```

```
---AlexSchroeder
#+END_VERSE
```

Quando quoti un passaggio da un'altro documento, é costume formattarlo come un paragrafo che é indentato in entrambi i margini sinistro e destro. Puoi mettere questa citazione in un documento Org nel modo seguente:

```
#+BEGIN_QUOTE
Everything should be made as simple as possible,
but not any simpler ---Albert Einstein
#+END_QUOTE
```

Se vuoi centrare del testo, devi fare in questo modo:

```
#+BEGIN_CENTER
Everything should be made as simple as possible, \\
but not any simpler
#+END_CENTER
```

12.2 Enfasi e mono-space

Puoi creare parole ***bold***, */italic/*, *_sottolineate_*, `=verbatim=` e `~codice~`, e, se non basta, **+barrato+**. Il testo nelle stringhe codice e verbatim non viene processato dalla specifica sintassi di Org; Viene esportato in verbatim.

12.3 L^AT_EX integrato

Per le note scientifiche che hanno bisogno di contenere simboli matematici e per le formule occasionali, Org mode supporta l'integrazione di codice L^AT_EX nei sui file. Puoi usare direttamente una sintassi simile al T_EX per simboli speciali, inserire formule e interi ambienti L^AT_EX.

The radius of the sun is $R_{\text{sun}} = 6.96 \times 10^8 \text{ m}$. On the other hand, the radius of Alpha Centauri is $R_{\{\text{Alpha Centauri}\}} = 1.28 \times R_{\{\text{sun}\}}$.

```
\begin{equation}                                % arbitrary environments,
x=\sqrt{b}                                       % even tables, figures
\end{equation}                                   % etc
```

If $a^2=b$ and $\wedge b=2 \wedge$, then the solution must be either $\$ a=\sqrt{2} \$$ or $\$ a=-\sqrt{2} \$$.

```
#+end_
```

12.4 Esempi letterali

Puoi inserire esempi letterali che non saranno soggetti al markup. Come gli esempi in mono-space, questi sono trattati come i codici sorgente ed esempi simili.

```
#+BEGIN_EXAMPLE
  Some example from a text file.
#+END_EXAMPLE
```

Per semplicità quando usi un piccolo esempio, puoi anche fare iniziare la linea dell'esempio con un due-punti seguito da uno spazio. Ci possono essere anche diversi spazi dopo il primo spazio vuoto che segue i due-punti:

```
Qui c'è un esempio
  : Qualche esempio per un file di testo.
```

Se l'esempio é del codice sorgente proveniente da un linguaggio di programmazione, o ogni altro tipo di testo che possa essere elaborato da Font Lock in Emacs, puoi chiedere che l'esempio sia mostrato tipo un buffer Emacs fontificato.

```
#+BEGIN_SRC emacs-lisp
  (defun org-xor (a b)
    "Exclusive or."
    (if a (not b) b))
#+END_SRC
```

Per modificare l'esempio in un buffer speciale che supporti questo linguaggio, usa C-' sia per entrare che per uscire dal buffer di modifica.

12.5 Immagini

Un'immagine é un link ad un file immagine che non ha la parte descrittiva, per esempio

```
./img/cat.jpg
```

Se desideri definire una descrizione per l'immagine e una label per un riferimento interno (vedi 5), assicurati che il link abbia una linea che lo precede con le parole chiave **CAPTION** e **NAME** come sotto:

```
#+CAPTION: This is the caption for the next figure link (or table)
#+NAME:    fig:SED-HR4049
[[./img/a.jpg]]
```

12.6 Creare note a piè di pagina

Una nota a piè di pagina è definita in un paragrafo che inizia con un marcatore di footnote tra parentesi quadrate nella colonna 0, non sono permesse indentazioni. I riferimenti alle note a piè di pagina sono dei semplici marcatori di parentesi quadrate, con dentro del testo. Per esempio:

```
The Org website[fn:1] now looks a lot better than it used to.
...
[fn:1] The link is: https://orgmode.org
```

I seguenti comandi gestiscono i footnote

C-c C-x f (org-footnote-action) Il comando azione dei footnote. Quando il punto è su un riferimento al footnote, salta alla definizione. Quando è in una definizione, salta al (primo) riferimento. Altrimenti, crea un nuovo footnote. Quando questo comando è chiamato con un argomento prefisso, un menu aggiuntivo di opzioni, incluso la rinumerazione, viene mostrato.

C-c C-c (org-ctrl-c-ctrl-c) Salta tra la definizione e il riferimento.

13 Esportazione

Org può convertire ed esportare i documenti in un varietà di altri formati che potresti ritenere meglio strutturati (vedi 3) e markup (vedi 12) se possibile.

13.1 Il dispatcher di esportazione

L'interfaccia di esportazione è l'interfaccia principale per l'esportazione Org. Un menu gerarchico presenta la configurazione corrente per i formati di esportazione. Le opzioni vengono mostrate come semplici bottoni nella stessa schermata.

C-c C-e (org-export-dispatcher) Richiama l'interfaccia di esportazione,

Org di default esporta l'intero buffer. Se il buffer Org ha una regione attiva, allora Org esporta solo quella regione.

13.2 Impostazioni di esportazione

L'esportatore riconosce delle linee speciali nel buffer che forniscono informazioni aggiuntive. Queste linee possono essere inserite in ogni punto del file:

```
#TITLE: I'm in the Mood for Org
```

La principali opzioni di esportazione includono:

TITLE	Il titolo che deve essere mostrato
AUTHOR	L'autore (di default viene da user-full-name)
DATE	una data, fissa, oppure un timestamp Org
EMAIL	email (di default viene da user-mail-address)
LANGUAGE	il codice di linguaggio, es. it

Il set di opzioni keyword possono essere impostate dal dispatcher (vedi 13.1) usando il comando **Insert template** premendo **#**

13.3 Tabelle dei contenuti

La tabella dei contenuti include tutte le intestazioni del documento. Va in profondità come sono i livelli delle intestazioni nel file. Se vuoi usare una differente profondità, o disabilitarlo per intero, imposta la variabile **org-export-with-toc**. Puoi fare la stessa cosa per ogni file, usando i seguenti elementi **toc** nella parola chiave **OPTIONS**:

```
#+OPTIONS: toc:2          (include solo due livelli nella TOC)
#+OPTIONS: toc:nil        (non esporta la TOC)
```

Org normalmente inserisce la tabella dei contenuti direttamente prima della prima intestazione nel file.

13.4 Includere file

Durante l'esportazione, puoi includere il contenuto di un'altro file. Per esempio, per includere il tuo file **.emacs**, puoi usare:

```
#+INCLUDE: "~/ .emacs" src emacs-lisp
```


Il primo parametro é il nome del file da includere. Il secondo parametro, opzionale, specifica il tipo di blocco **example**, **export** o **src**. Il terzo parametro, anch'esso opzionale, specifica il linguaggio del codice sorgente da usare per formattare il contenuto. Questo é rilevante sia per il tipo di blocchi **export** che **src**.

Puoi vedere il file incluso con C-c '.

13.5 Linee di commento

Le linee che iniziano con zero o più spazi bianchi seguiti da un **#** e da uno spazio bianco sono trattati come commenti e, come tali, non vengono esportati.

Come del resto, le zone circondate da **#+BEGIN_COMMENT . . . #+END_COMMENT** non sono esportate.

In fine, una keyword **COMMENT** all'inizio di un elemento, ma prima di un'altra parola chiave o cookie, commenta l'intero sotto-albero. Il comando sotto aiuta a cambiare lo stato di commento di una intestazione.

C-c ; (org-toggle-comment) Attiva/Disattiva la parola chiave **COMMENT** all'inizio di un elemento

13.6 Esportazione ASCII/UTF-8

L'esportazione ASCII produce in output un file di soli caratteri ASCII. Questo é il più semplice e diretto testo di output. Non contiene nessun markup Org. L'esportazione UTF-8 usa caratteri aggiuntivi e simboli avviabili in questo standard di codifica.

C-c C-e t a, C-c C-e t u (org-ascii-export-to-ascii) Esporta come file ASCII con estensione **.txt**. Con **myfile.org** Org esporta **myfile.txt**, riscrivendolo senza warning. Per **myfile.txt**, Org esporta in **myfile.txt.txt** per prevenire il rischio di perdita dei dati.

13.7 Esportazione HTML

La modalità Org contiene un esportatore in HTML con estensione HTML e formatta compatibile con XHTML 1.0 strict standard.

C-c C-e h h (org-html-export-to-html) Esporta con file HTML con estensione **.html**. Per **myfile.org**, Org lo esporta come **myfile.html**, sovrascrivendolo senza warning. **C-c C-e h o** esporta in HTML e apre il file nel browser web.

Il backend di esportazione trasforma le `< e >` in `<` e `>`. Per inserire del puro codice HTML nel file Org in modo che il backend di esportazione possa includerlo nell'output, usa questa sintassi in linea: `@@html:...@@`. Per esempio:

```
@@html:<b>@@bold text@@html:</b>@@
```

Per un blocco di codice HTML puro più ampio usa questo blocco di codice di esportazione:

```
#+HTML: Literal HTML code for export
```

```
#+BEGIN_EXPORT html
```

Tutte le linee tra questi marker sono esportati letteralmente.

```
#+END_EXPORT
```

13.8 Esportazione \LaTeX

L'esportazione \LaTeX può elaborare documenti complessi, incorporando classi \LaTeX standard o personalizzate, genera documenti usando engine \LaTeX alternativi, e producendo un file PDF con indici, bibliografie e tabelle dei contenuti, destinato alla visione interattiva o per la pubblicazione di stampa di alta qualità.

Di default, l'output \LaTeX usa la classe *article*. Puoi cambiarlo aggiungendo, nel tuo file, un'opzione tipo questa `#+LATEX_CLASS: myclass`. Le classi possono essere inserite in `org-latex-classes`.

C-c c-e l l (`org-latex-export-to-latex`) Esporta in un file \LaTeX con estensione `.tex`. Per `myfile.org`, Org lo esporta in `myfile.tex`, sovrascrivendo senza warning.

C-c C-e l p (`org-latex-esport-to-pdf`) Esporta come file \LaTeX e lo converte in file PDF.

C-c C-e l o (`<no corresponding named command>`) Esporta come file \LaTeX e lo converte in PDF, dopo di che apre il file PDF usando il visualizzatore di default.

Il backend di esportazione \LaTeX può inserire codice \LaTeX arbitrario, vedi 12.3. Ci sono tre modi per inserire del codice nel file Org e ognuno usa una sintassi di quotazione diversa.

Inserimento in linea quota con i simboli @:

Code embedded in-line `@@latex:any arbitrary LaTeX code@@` in a paragraph.

Inserimento di una o più linee con keyword nel file Org:

```
#+LATEX: any arbitrary LaTeX code
```

Inserimento di un blocco di esportazione nel file Org, dove il backend esporterà tutto il codice tra il marcatore di inizio e fine:

```
#+BEGIN_EXPORT latex
    any arbitrary LaTeX code
#+END_EXPORT
```

13.9 Esportazione iCalendar

L'enorme successo dell'interoperabilità della modalità Org é l'abilità di esportare ed importare da applicazioni esterne. Il backend di esportazione iCalendar prende le date di calendario dai file Org e le esporta nel formato standard iCalendar.

C-c C-e c f (org-icalendar-export-to-ics) Crea un elemento iCalendar dal buffer Org corrente e lo memorizza nella stessa directory, usando un file con estensione `.ics`.

C-c C-e c c (org-icalendar-combine-agenda-files) Crea un file iCalendar combinando i file Org provenienti da `org-agenda-files` e scrivendoli nel nome di file `org-icalendar-combined-agenda-file`.

14 Pubblicare

Org include un sistema di gestione delle pubblicazione che ti permette di configurare una conversione HTML automatica dei *progetti* composta dai file Org collegati internamente. Puoi anche configurare Org in modo che carichi in automatico le pagine HTML e relativi allegati, come immagini e file di codice sorgente, in un server web.

Puoi anche usare Org per convertire i file in PDF, o anche combinando la conversione HTML e PDF in modo che questi file siano disponibili in entrambi i formati sul server.

Per delle istruzioni dettagliate sulle impostazioni, vedi `manual`. Qui c'è un esempio:

```
(setq org-publish-project-alist
  '(("org"
    :base-directory "~/org/"
    :publishing-function org-html-publish-to-html
    :publishing-directory "~/public_html"
    :section-numbers nil
    :with-toc nil
    :html-head "<link rel=\"stylesheet\"
               href=\"../other/mystyle.css\"
               type=\"text/css\"/>"))))
```

C-c C-e P x (org-publish) Mostra un prompt per uno specifico progetto e pubblica tutti i file che lo riguardano.

C-c C-e P p (org-publish-current-project) Pubblica il progetto che contiene il file corrente.

C-c C-e P f (org-publish-current-file) Pubblica solo il file corrente.

C-c C-e P a (org-publish-all) Pubblica tutti i progetti.

Org usa il timestamp per tenere traccia di quando un file viene cambiato. Le funziona sopra normalmente pubblicano solo i file modificati. Puoi bypassare questo e forzare la pubblicazione di tutti i file dando un argomento prefisso ad ogni comando visto sopra.

15 Lavorare con il codice sorgente

Org mode fornisce un numero di funzionalità per lavorare con i codici sorgenti, incluso la modifica di blocchi di codice nella loro major-mode, valutazione del blocco di codice, tangling di blocchi di codice ed esportazione di blocchi di codice e del loro risultato in diversi formati.

Un blocco di codice si forma con questa struttura:

```
#+NAME: <name>
#+BEGIN_SRC <language> <switches> <header arguments>
  <body>
#+END_SRC
```

dove:

- `<name>` é una stringa usata per dare un nome univoco al blocco di codice,
- `<language>` specifica il linguaggio del blocco di codice, es. `emacs_lisp`, `shell`, `R`, `python` ecc..
- `<switches>` può essere usato per controllare l'esportazione del blocco di codice
- `<header arguments>` può essere usato per controllare diversi aspetti del comportamento del blocco di codice come dimostrato sotto
- `<body>` contiene il codice sorgente.

Usa `C-c` ' per modificare il blocco di codice corrente. Apre un nuovo buffer di modifica nella `major-mod` contenente il sorgente del blocco di codice, pronto per la modifica. Usando `C-c` ' nuovamente chiude il buffer e torna al buffer `Org`.

Usare argomenti di intestazione

Un argomento di intestazione viene specificato con un carattere due-punti iniziale seguito dal nome dall'argomento in minuscolo.

Gli argomenti di intestazione possono essere impostati in diversi modi; `Org` gli assegna delle priorità per evitare che ci siano delle sovrapposizioni o dei conflitti dando alle impostazioni locali la priorità più alta.

Argomenti di intestazione a livello di sistema Questi sono specificati personalizzando la variabile `org-babel-default-header-args`, o, per uno specifico linguaggio `LANG` `org-babel-default-header-args:LANG`.

Argomenti di intestazione nelle proprietà Puoi impostarlo usando la proprietà `header-args` (vedi 8)—o `header-args:LANG` per linguaggio `LANG`. Gli argomenti di intestazione nelle proprietà vengono applicati nei sotto-alberi dei livelli sotto.

Argomenti di intestazione nei blocchi di codice Gli argomenti di intestazione sono comunemente usati per impostare il sorgente dei blocchi di codice, nella linea `BEGIN_SRC`:

```
#+NAME: factorial
#+BEGIN_SRC haskell :results silent :exports code :var n=0
```

```

    fac 0 = 1
    fac n = n * fac (n-1)
#+END_SRC

```

Gli argomenti di intestazione nei blocchi di codice possono essere su più linee usando la parola chiave **HEADER** su ogni linea.

Valutare i blocchi di codice

Usa C-c C-c per valutare il blocco di codice corrente ed inserire il risultato nel documento Org. Di default, é abilitata solo la valutazione dei blocchi di codice **emacs-lisp**, comunque esiste il supporto per la valutazione di blocchi di codice in diversi linguaggi. Per una lista completa dei linguaggi supportati guarda il manuale. Di seguito viene mostrato un blocco di codice ed il suo risultato.

```

#+BEGIN_SRC emacs-lisp
  (+ 1 2 3 4)
#+END_SRC

```

```

#+RESULTS:
: 10

```

La sintassi qui sotto viene usata per passare un argomento al blocco di codice usando l'argomento di intestazione **var**.

```

:var NAME=ASSIGN

```

NAME é il nome della variabile usata nel corpo del blocco di codice. ASSIGN é il valore letterale, come una stringa, un numero, un riferimento ad una tabella, una lista, un esempio letterale, un'altro blocco di codice—con o senza argomenti —o il risultato della valutazione di un blocco di codice.

Risultato della valutazione

Il modo in cui Org gestisce i risultati dell'esecuzione di un blocco di codice dipende da come i molti argomenti dell'intestazione lavorano assieme. Il fattore determinante, tuttavia, è l'argomento dell'intestazione **result**. Lui controlla la *collezione*, *tipo*, *formato* e la *manipolazione* dei risultati dei blocchi di codice.

Collezione Come il risultato debba essere collezionato dal blocco di codice.

Puoi scegliere tra `output` o `value` (il default).

Tipo Che tipo di risultato mi aspetto dall'esecuzione del blocco di codice.

Puoi scegliere tra `table`, `list`, `scalar` e `file`. Org prova ad indovinare se non lo hai impostato.

Formato Come Org processa il risultato. Alcuni possibili valori sono `code`, `drawer`, `html`, `latex`, `link` e `raw`.

Manipolazione Come viene inserito il risultato una volta formattato appropriatamente. I valori consentiti sono `silent`, `replace` (il default), `append` o `prepend`.

I blocchi di codice con i risultati di output in file—es.: grafici, diagrammi e figure—possono accettare un argomento di intestazione `:file FILENAME`, in questo caso i risultati sono salvati nel file nominato, e un link al file viene inserito nel buffer.

Esportazione dei blocchi di codice

È possibile esportare il *codice* del blocco di codice, il *risultato* della valutazione del blocco di codice, *entrambe* il codice ed il risultato della valutazione del blocco di codice, o *niente*. Org di default, per diversi linguaggi, esporta il *codice*.

L'argomento d'intestazione `exports` serve ad informare che quella parte di codice viene esportata, per esempio, nel formato HTML o \LaTeX . Può essere impostato a `code`, `results`, `both` o `none`.

Estrazione del codice sorgente

Usa `C-c C-v t` per creare un file con solo il codice sorgente estratto dal blocco di codice del buffer corrente. Questo viene definito come “tangling”—un termine adottato dalla comunità di programmazione letterale. Durante il tangling del blocco di codice il suo corpo viene espanso usando `org-babel-expand-src-block`, che può espandere sia i riferimenti di tipo variabile che quelli di tipo “Noweb”. Per fare il tangle di un blocco di codice, il blocco, deve avere un argomento di intestazione `tangle`, vedi il manuale per maggiori dettagli.

16 Miscellanee

Completion

Org ha il completamento nel buffer con M-TAB. Il mini-buffer non é coinvolto. Digita una o più lettere per poi usare la combinazione di scelta rapida per completare il testo sul posto.

Per esempio, questo comando completerà i simboli Tex dopo \, le keyword TODO all’inizio di una intestazione e i tag dopo i : nelle intestazioni.

Modelli di struttura

Per inserire velocemente dei blocchi strutturati vuoti, come `+BEGIN_SRC ...` `#+END_SRC`, o per racchiudere il testo in un blocco, usa:

C-c C-, (`org-insert-structure-template`) Mostra un prompt per i tipi di strutture a blocchi, e inserisce il blocco nel punto. Se la regione é attiva, la racchiude nel blocco.

Viste pulite

L’outline di default di Org con asterischi e senza indentazione può andare bene per piccoli documenti. Per lunghi documenti come *libri*, l’effetto non é evidente. Org fornisce un’alternativa per gli asterischi e uno schema di indentazione, come mostrato alla destra della seguente tabella. Usa un solo asterisco e indenta il testo per allinearlo all’intestazione:

* Top level headline		* Top level headline
** Second level		* Second level
*** Third level		* Third level
some text		some text
*** Third level		* Third level
more text		more text
* Another top level headline		* Another top level headline

Questo tipo di visualizzazione può essere ottenuta dinamicamente sul display usando la modalità Org Indent (M-x `org-indent-mode` RET), che inserisce uno spazio intangibile su ogni riga. Puoi attivare la modalità Org Indent per tutti i file personalizzando la variabile `org-startup-indented`, oppure puoi attivarlo in ogni file usando:

```
#+STARTUP: indent
```


Se si desidera che l'indentazione sia costituita da caratteri di spazio fisso, in modo che il file di testo semplice appaia il più possibile simile alla visualizzazione di Emacs. Org aiuta a rientrare (con TAB) il testo sotto ad ogni suo titolo, a nascondere le stelle iniziali e a usare solo i livelli 1, 3, ecc., ottenendo un rientro di due caratteri per ogni livello. Per ottenere questo in un file usa

```
#+STARTUP: hidestarts odd
```

17 Traduzione

Il titolo originale di questo documento é **Org Mode Compact Guide** che potete trovare all'indirizzo <https://orgmode.org/orgguide.html>, la mia é una traduzione non ufficiale di tale documento.