



GUIDA ALLA CLASSE
GUIDATEMATICA



INTRODUZIONE

Questa breve guida serve per descrivere la classe *guidatematica* predisposta per comporre le guide tematiche del $\text{\textcolor{teal}{G}\text{\textcolor{teal}{I}\text{\textcolor{teal}{T}}}}$ con uno stile comune, in modo che già il solo vedere il frontespizio permetta di riconoscere le guide tematiche dalle altre guide e dalle altre pubblicazioni del $\text{\textcolor{teal}{G}\text{\textcolor{teal}{I}\text{\textcolor{teal}{T}}}}$.

La classe *guidatematica* è dotata del file `.dtx` e la sua documentazione è ampliata con questo documento. Sarebbe meglio che nessuno modificasse il file, altrimenti si perderebbe lo scopo per il quale è stato scritto, cioè di rendere omogenee nello stile le guide tematiche del $\text{\textcolor{teal}{G}\text{\textcolor{teal}{I}\text{\textcolor{teal}{T}}}}$. Se si desiderano proporre estensioni, cambiamenti, altre funzionalità, ci si deve rivolgere ai responsabili del $\text{\textcolor{teal}{G}\text{\textcolor{teal}{I}\text{\textcolor{teal}{T}}}}$ che curano il sito Web e le pubblicazioni del Gruppo.

Questo stesso documento è stato composto con la classe *guidatematica*.

Questa classe è costruita sulla classe *memoir* quindi l'utente può fare riferimento a tutte le funzionalità di quella classe, tranne alcune che collidono con i pacchetti già caricati; se decide di comporre l'indice analitico legga con attenzione l'apposito capitolo (vedi pag. 35); se decide di modificare lo stile di alcune liste o di crearne di nuove faccia riferimento alle funzionalità del pacchetto *enumitem* e alla sua documentazione.

Questa documentazione si riferisce alla classe *guidatematica.cls*, versione v.0.6c, del 2012/12/12.

Essa è stata composta con $\text{\textcolor{teal}{X}\text{\textcolor{teal}{L}\text{\textcolor{teal}{A}\text{\textcolor{teal}{T}\text{\textcolor{teal}{E}\text{\textcolor{teal}{X}}}}}}$ usando i font Latin Modern Opentype forniti insieme alla distribuzione di TeX Live 2012 e accessibili con i comandi propri di $\text{\textcolor{teal}{X}\text{\textcolor{teal}{L}\text{\textcolor{teal}{A}\text{\textcolor{teal}{T}\text{\textcolor{teal}{E}\text{\textcolor{teal}{X}}}}}}$.

Questa documentazione è soggetta alla licenza L^AT_EX Project Public Licence, versione 1.3 o successive; il testo della licenza è sempre contenuto in qualunque distribuzione del sistema T_EX, e nel sito <http://www.latex-project.org/lppl.txt>.

RINGRAZIAMENTI

Io ho scritto materialmente questa classe *guidatematica* cercando di mettermi nella veste del curatore di un lavoro collettivo del $\mathcal{G}_I\mathcal{T}$ e per il $\mathcal{G}_I\mathcal{T}$; ho consultato quindi gli autori o coautori delle guide tematiche pubblicate finora, Claudio Fiandrino, Roberto Giacomelli e Tommaso Gordini, che ringrazio moltissimo del loro contributo, delle loro osservazioni, dei loro suggerimenti; questo è il miglior modo per contribuire al $\mathcal{G}_I\mathcal{T}$, e quindi siamo tutti autori a pari merito di questa classe; se ne aggiungeranno altri col tempo e verranno aggiunti a questa breve lista dei “fondatori”.

CLAUDIO BECCARI

claudio dot beccari at gmail dot com

INDICE

INTRODUZIONE	2
RINGRAZIAMENTI	3
INDICE	4
1 GUIDA RAPIDA ALLA CLASSE	5
1.1 Installazione della classe	5
1.2 Uso della classe	6
2 DOCUMENTAZIONE DELLA CLASSE	18
2.1 Pacchetti precaricati	19
2.2 Nuovi comandi	23
2.3 Nuovi ambienti	27
2.4 La virgola intelligente	32
3 GLI INDICI ANALITICI	35
4 COMPORRE CON pdfLaTeX O CON XeLaTeX	38
BIBLIOGRAFIA	42
INDICE ANALITICO	44

GUIDA RAPIDA ALLA CLASSE

1

1.1 INSTALLAZIONE DELLA CLASSE

La classe viene distribuita in un kit da installare ed usare come quello distribuito per $\text{\texttt{Ar\TeX}}\text{nica}$.

La via più spiccia è quella di copiare il kit `guidatematica.zip` nella cartella dove risiede il file sorgente (unico o principale) con cui si vuole comporre la propria guida tematica; in quella cartella si decompime il kit e tutti i file, compresi quelli della documentazione, risiedono in quella stessa cartella e sono immediatamente usabili per la composizione *solo* di quella guida tematica.

Meglio sarebbe disporre dell'albero personale di $\text{\texttt{T\TeX}}$; se non se ne dispone, lo si crea secondo le modalità descritte per il proprio sistema operativo e la propria distribuzione del sistema $\text{\texttt{T\TeX}}$. Se se ne dispone già, basta creare le cartelle a cui si fa riferimento qui di seguito. Si decompime il kit in una cartella qualsiasi, meglio se in¹ `$HOME/texmf/source/latex/-guidatematica/`. Poi si spostano i file `GuidaTematica-doc.pdf` e `guidatematica.pdf` nella cartella `$HOME/texmf/doc/latex/guidatematica/` e gli altri file in `$HOME/texmf/tex/latex/guidatematica/`, tranne il file `guidatematica.dtx` che resta nella cartella dove si è eseguita la decompressione. Se la distribuzione del sistema $\text{\texttt{T\TeX}}$ lo richiede anche per l'albero personale, bisogna ricordarsi di creare o di aggiornare il database dei nomi dei file di questo albero usando gli strumenti della propria distribuzione. A questo punto tutti i programmi di composizione del sistema $\text{\texttt{T\TeX}}$ sono

¹`$HOME` indica la propria cartella che generalmente ha il nome dell'utente, ma dipende molto dal sistema operativo; qui si è usato per il separatore delle cartelle la barra dritta, come è normale per le macchine di tipo UNIX; per le macchine Windows basta sostituire la barra dritta con la barra rovescia.

in grado di trovare i file necessari per comporre *qualsiasi* guida tematica, qualunque sia la cartella dove i suoi file sorgente risiedono.

1.2 USO DELLA CLASSE

Lo schema generale da usare per comporre una guida tematica è il seguente:

```
\documentclass[b5paper,11pt,openany,ipertesto]{guidatematica}
\ProvidesFile{⟨nome del file⟩}[⟨aaaa/mm/gg v.x.y Descrizione⟩]
```

```
⟨qualche definizione o qualche ulteriore pacchetto⟩
⟨eventuali impostazioni per \makeindex⟩
```

```
\begin{document}
```

```
\frontmatter
```

```
\author{⟨autore o autori⟩}
```

```
\title{⟨titolo⟩}
```

```
\maketitle
```

```
⟨eventuale pagina legale⟩
```

```
\chapter{⟨titolo dell'introduzione⟩}
```

```
⟨corpo dell'introduzione⟩
```

```
\newpage\tableofcontents
```

```
\mainmatter
```

```
\chapter{⟨titolo del capitolo⟩}
```

```
⟨corpo del capitolo⟩
```

```
...
```

```
\backmatter
```

```
\chapter{⟨titolo della "postfazione"⟩}
```

```
⟨corpo dell'eventuale postfazione⟩
```

⟨eventuale ridefinizione di `\prebibhook` per fare precedere un breve testo esplicativo all’elenco bibliografico vero e proprio⟩

```
\bibliography{⟨nome del database bibliografico senza estensione⟩}
```

```
%%% eventuali indici analitici
```

```
\printindex
```

```
\printindex[⟨nome dell’indice⟩]
```

```
...
```

```
\end{document}
```

OSSERVAZIONI

Il testo precedente, benché scritto in un file PDF, può essere selezionato, copiato e incollato in un file `.tex` e usato come modello (o come *template* come è venuto di moda chiamarlo) per iniziare a strutturare il file sorgente di una guida tematica.

Comunque vale la pena di illustrare alcuni punti. Nel farlo, ci si rivolge a persone che, avendo qualcosa da dire in merito a qualche sfaccettatura del “sistema \TeX ”, hanno sicuramente esperienza nello scrivere documenti usando il mark-up \LaTeX , e conoscono molto bene pregi e difetti dei programmi di composizione \pdfLaTeX e \XeLaTeX . Quindi molte cose saranno date per scontate.

INTRODUZIONE questo capitolo non numerato scritto nella “front matter” può avere un titolo diverso da “Introduzione”. Non è nemmeno necessario scrivere un’introduzione, prefazione, presentazione, comunque la si voglia chiamare, ma si ritiene che un minimo di spiegazioni o motivazioni per avere scritto una guida tematica sia utile per il lettore.

Alla fine dell’introduzione ci sta bene qualche cenno alla licenza, indipendentemente da quello che si è scritto nella *⟨eventuale pagina legale⟩*. Ci sta bene qualche indicazione in più e forse anche il motivo per avere scelto questa piuttosto che quella licenza. Qualunque licenza si scelga deve essere una licenza libera, altrimenti il \GJIT non può pubblicare la guida.

Spesso la fine dell’introduzione è il posto adatto per inserire il nome dell’autore ed eventualmente anche l’indirizzo di posta elettronica; l’utente

esperto non ha nessun problema ad usare una *minipage* di larghezza pari ad una frazione della `\textwidth` con il suo contenuto centrato e con l'indirizzo scritto nella forma `mario dot rossi at server dot network`; naturalmente non è vietato che l'autore scriva `mario.rossi@server.network`, ma è più prudente la prima forma che risulta più difficile da carpire da parte di malware vari.

PREAMBOLO il preambolo in generale può essere vuoto; può succedere però che l'utente necessiti di qualche definizione particolare che non si trova in nessun pacchetto o che abbia bisogno di ulteriori pacchetti oltre quelli già caricati (si veda più avanti l'elenco di questi pacchetti). In ogni caso, se desidera comporre la sua guida tematica in modo da usare collegamenti ipertestuali, deve egli stesso specificare l'opzione *ipertesto* fra le opzioni della classe; questa opzione dilaziona la chiamata al pacchetto *hyperref* con alcune personalizzazioni al momento di completare il comando `\begin{document}`; il pacchetto viene caricato per ultimo perché questa operazione deve essere fatta dopo il caricamento dei pacchetti personali. D'altronde i collegamenti ipertestuali non sono obbligatori, ma sono utili per la lettura al monitor solo per alcune guide e non per altre; è una scelta dell'utente quella di voler usare i collegamenti ipertestuali; è quindi una scelta dell'utente quella di specificare l'opzione suddetta.

INFORMAZIONI PER LA PAGINA DEL TITOLO Potrebbe essere interessante usare il comando `\date` per inserire nella pagina del titolo informazioni relative alla versione del documento. Per non dover ripetere sempre le stesse cose scrivendole varie volte, è opportuno scrivere subito dopo la specificazione della classe da usare con le sue opzioni, la riga che comincia con `\ProvidesFile`; il *<nome del file>* sarà il nome del file sorgente principale della guida tematica; fra le informazioni facoltative (che in questo caso non sono facoltative affatto) *aaaa/mm/gg* indica la data di ultima modifica del file; *v.x.y* indica la versione *x* e subversione *y*; *Descrizione* fornisce con poche parole la descrizione della guida.

Successivamente nel predisporre le informazioni che `\maketitle` userà per comporre il frontespizio si potrà scrivere:


```

\author{\langle autore o autori \rangle}
\title{\langle titolo \rangle}
\subtitle{\langle eventuale sottotitolo \rangle}
\GetFileInfo{\langle nome del file \rangle}
\date{\filename\ \fileversion\ del \filedate}
\maketitle

```

ripetendo il $\langle \text{nome del file} \rangle$ completo di estensione solo nell'argomento di `\GetFileInfo`, ma senza bisogno di riscrivere i dati ad ogni aggiornamento del file stesso.

PAGINA LEGALE Si tenga presente che la classe *guidatematica* è rilasciata con la licenza LPPL, L^AT_EX Project Public Licence, reperibile nel sito <http://www.latex-project.org/lppl.txt>. L'autore di una guida tematica della collana del G_UI_T può anche decidere di usare un'altra licenza, come per esempio quella della Creative Commons, e indicare le clausole della licenza con le apposite icone. Le possibili combinazioni di licenze e di mutue interazioni rendono praticamente impossibile creare una macro o un ambiente per specificare quanto la legge impone a seconda delle licenze invocate. Si suggerisce di usare la licenza LPPL anche per le guide tematiche, ma sta comunque all'autore il compito di scrivere la paginetta indicata nel modello con $\langle \text{eventuale pagina legale} \rangle$.

SEZIONAMENTI I comandi `\frontmatter`, `\mainmatter` e `\backmatter` sono obbligatori, ma tocca all'utente indicarli nei posti opportuni.

POSTFAZIONE La postfazione, qualunque titolo essa abbia, è evidentemente facoltativa e non è assolutamente necessario scriverne una solo perché c'è la possibilità di farlo; d'altronde, se si è abituati a scrivere una breve sezione introduttiva alla lettura della bibliografia, non ci si dimentichi che la bibliografia consente di comporre un breve testo prima dell'elenco dei riferimenti, come spiegato qui sotto.

BREVE TESTO INTRODUTTIVO ALLA BIBLIOGRAFIA Per comporre un breve testo da comporre fra il titolo della bibliografia e l'elenco bibliografico vero e proprio, bisogna passarlo come argomento al comando `\prebibhook`; siccome questo comando è predefinito per contenere una stringa nulla, il breve testo deve venire assegnato a `\prebibhook` ridefinendolo mediante `\renewcommand`.

UNO O PIÙ INDICI ANALITICI L'indice analitico non è quasi mai necessario; più indici analitici ancor meno. Ma l'utente che voglia comporre uno o più indici analitici è reindirizzato alla documentazione del pacchetto *imageidx* che consente di farlo nella maniera più comoda finora sviluppata per la loro composizione.

Tabella 1.1: Sinossi dei comandi e ambienti specifici della classe

Entità	Descrizione
<code>\meta</code>	per definire un argomento <code>\meta{argomento} → ⟨argomento⟩</code>
<code>\marg</code>	per definire un argomento obbligatorio <code>\marg{argomento} → {⟨argomento⟩}</code>
<code>\oarg</code>	per definire un argomento facoltativo <code>\oarg{argomento} → [⟨argomento⟩]</code>
<code>\garg</code>	per definire le coordinate per un ambiente grafico <code>\garg{x,y} → (⟨x⟩,⟨y⟩)</code>
<code>\comando</code>	per comporre un comando senza eseguirlo <code>\comando{\comando} → \comando</code>
<code>\cs</code>	per comporre un comando usandone solo il nome, inviandone eventualmente le informazioni necessarie all'indice analitico <code>\cs{comando} → \comando</code>
<code>\Bambiente</code>	per comporre l'apertura di un ambiente specifico <code>\Bambiente{ambiente} → \begin{ambiente}</code>
<code>\Eambiente</code>	per comporre la chiusura di un ambiente specifico <code>\Eambiente{ambiente} → \end{ambiente}</code>
<code>\Sambiente</code>	per comporre la sintassi dell'apertura di un ambiente con un solo argomento opzionale <code>\Sambiente{ambiente}{N.arg.} → \begin{⟨ambiente⟩}[⟨N.arg.⟩] ... \end{⟨ambiente⟩}</code>
<code>\Dambiente</code>	per comporre la sintassi dell'apertura di un ambiente con due argomenti opzionali <code>\Dambiente{ambiente}{N.arg.}{default} → \begin{⟨ambiente⟩}[⟨N.arg.⟩][⟨default⟩] ... \end{⟨ambiente⟩}</code>

(continua nella pagina successiva)

Sinossi dei comandi e ambienti specifici della classe (*continua*)

<code>\Arg</code>	per comporre un argomento specifico racchiuso fra graffe <code>\Arg{argomento} → {argomento}</code>
<code>\GetFileInfo</code>	Per ricavare le informazioni relative ad un file provvisto al suo inizio del comando <code>\Provides...: \GetFileInfo{book.cls}</code> ; non produce risultati compositivi, ma assegna nuovi valori alle variabili <code>\filename</code> , <code>\fileversion</code> e <code>\filedate</code>
<code>\setcopymark</code>	scrive in verticale nel margine esterno quello che si desidera <code>\setcopymark{Composizione del \today}</code> → vedi margine esterno
<code>\maketitle</code>	comando ridefinito in modo che componga la pagina del titolo in modo uniforme fra tutte le guide tematiche della collezione
<code>\subtitle</code>	ulteriore comando per specificare un sottotitolo da comporre nella pagina del titolo: <code>\subtitle{Sottotitolo della guida}</code>
<code>\makeindex</code>	comando modificato per l'uso del pacchetto <i>imakeidx</i>
<code>\showindexmarks</code>	mostra le parole indicizzate nella parte superiore del margine esterno della pagina; talvolta fallisce se le corrispondenti stringhe sono troppo lunghe;
<code>\hideindexmarks</code>	nasconde l'indicazione delle parole indicizzate
<code>\textttm</code>	comando testuale per comporre col font typewriter monospaziato anche quando fosse in vigore quello proporzionale

(*continua nella pagina successiva*)

Sinossi dei comandi e ambienti specifici della classe (*continua*)

<code>\textttv</code>	comando testuale per comporre col font typewriter proporzionale anche quando fosse in vigore quello monospaziato
<code>\tvfamily</code>	dichiarazione corrispondente al comando <code>\textttv</code>
<code>\textl</code>	comando per comporre con i font typewriter con le cifre “maiuscole” anche quando fosse in vigore lo stile <code>OldStyle</code> per le cifre <code>\textl{123}</code> → 123 a confronto con 123 → 123
<code>\prebibhook</code>	comando per inserire un (breve) testo fra il titolo della bibliografia e l’elenco bibliografico
<code>\ambstyle</code>	comando per comporre il nome degli ambienti con un certo stile di scrittura <code>\ambstyle{ambiente}</code> → <i>ambiente</i>
<code>\classstyle</code>	comando per comporre il nome delle classi con un certo stile di scrittura <code>\classstyle{classe}</code> → <i>classe</i>
<code>\filestyle</code>	comando per comporre il nome dei file con un certo stile di scrittura <code>\filestyle{file}</code> → <i>file</i>
<code>\packstyle</code>	comando per comporre il nome dei pacchetti con un certo stile di scrittura <code>\packstyle{pacchetto}</code> → <i>pacchetto</i>
<code>\progstyle</code>	comando per comporre il nome dei programmi con un certo stile di scrittura <code>\progstyle{programma}</code> → <i>programma</i>
<code>\opzstyle</code>	comando per comporre il nome delle opzioni con un certo stile di scrittura <code>\opzstyle{opzione}</code> → <i>opzione</i>

(*continua nella pagina successiva*)

Sinossi dei comandi e ambienti specifici della classe (*continua*)

<code>\contastyle</code>	comando per comporre il nome dei contatori con un certo stile di scrittura <code>\contastyle{contatore} → contatore</code>
<code>\stilestyle</code>	comando per comporre il nome degli stili di pagina con un certo stile di scrittura <code>\stilestyle{stile} → stile</code>
<code>\numeristyle</code>	comando per comporre il nome degli stili di numerazione con un certo stile di scrittura <code>\numeristyle{arabic} → arabic</code>
<code>\umisurastyle</code>	comando per comporre il nome delle unità di misura con un certo stile di scrittura <code>\umisurastyle{mm} → mm</code>
<code>\chiavestyle</code>	comando per comporre il nome delle chiavi nella sintassi $\langle chiave \rangle = \langle valore \rangle$ con un certo stile di scrittura <code>\chiavestyle{angle} → angle</code>
<code>\descrittorestyle</code>	comando per comporre il codice dei descrittori delle colonne con un certo stile di scrittura <code>\descrittorestyle{c} → c</code>
<code>\posizionestyle</code>	comando per comporre il codice dei posizionamenti degli oggetti mobili con un certo stile di scrittura <code>\posizionestyle{t} → t</code>
<code>\allineamentostyle</code>	comando per comporre il codice degli allineamenti degli ambienti <i>tabular</i> , <i>array</i> e simili con un certo stile di scrittura <code>\allineamentostyle{b} → b</code>
<code>\opz</code>	compone una opzione nello stile <code>\opzstyle</code> e manda le informazioni all'indice analitico sotto la voce "opzione"
<code>\conta</code>	compone il nome di un contatore nello stile <code>\contastyle</code> e manda le informazioni all'indice analitico sotto la voce "contatore"

(*continua nella pagina successiva*)

Sinossi dei comandi e ambienti specifici della classe (*continua*)

<code>\stile</code>	compone il nome di uno stile di pagina nello stile <code>\stilestyle</code> e manda le informazioni all'indice analitico sotto la voce "stile della pagina"
<code>\numeri</code>	compone il nome di uno stile di numerazione nello stile <code>\numeristyle</code> e manda le informazioni all'indice analitico sotto la voce "numerazione"
<code>\misura</code>	compone il nome di una unità di misura nello stile <code>\umisurastyle</code> e manda le informazioni all'indice analitico sotto la voce "unità di misura"
<code>\chiave</code>	compone il nome di una chiave nello stile <code>\chiavestyle</code> e manda le informazioni all'indice analitico sotto la voce "chiave"
<code>\descrittore</code>	compone il codice di un descrittore di colonna nello stile <code>\descrittorestyle</code> e manda le informazioni all'indice analitico sotto la voce "descrittore di colonna"
<code>\posizione</code>	compone il codice di posizionamento di un oggetto mobile nello stile <code>\posizionestyle</code> e manda le informazioni all'indice analitico sotto la voce "posizione degli oggetti flottanti"
<code>\allineamento</code>	compone il codice di allineamento di una tabella o di una matrice nello stile <code>\allineamentostyle</code> e manda le informazioni all'indice analitico sotto la voce "codice di allineamento"
<code>\amb</code>	compone il nome di un ambiente nello stile <code>\ambstyle</code> e manda le informazioni all'indice analitico sotto la voce "ambiente"

(*continua nella pagina successiva*)

Sinossi dei comandi e ambienti specifici della classe (*continua*)

<code>\class</code>	compone il nome di una classe nello stile <code>\classstyle</code> e manda le informazioni all'indice analitico sotto la voce "classe"
<code>\file</code>	compone il nome di un file (con la sua estensione) oppure la sola estensione di un file composti nello stile <code>\filestyle</code> e manda le informazioni all'indice analitico sotto la voce "file"
<code>\pack</code>	compone il nome di un pacchetto nello stile <code>\packstyle</code> e manda le informazioni all'indice analitico sotto la voce "pacchetto"
<code>\prog</code>	compone il nome di un programma nello stile <code>\progstyle</code> e manda le informazioni all'indice analitico sotto la voce "programma"
<i>description</i>	si riferisce al solito ambiente di \LaTeX salvo che l'etichetta della voce da descrivere è composta in maiuscoletto
<i>blockdescription</i>	si riferisce ad un ambiente simile a <i>description</i> privo però del rientro del margine sinistro; l'etichetta è composta in maiuscoletto
<i>plaindescription</i>	si riferisce ad un ambiente simile a <i>description</i> con il rientro del margine sinistro; l'etichetta è composta in font normale, non in neretto come nell'ambiente descrittivo di \LaTeX
<i>compactitemize</i>	si riferisce ad una lista semplice (puntata) priva di spazi di separazione fra una voce e la successiva
<i>compactenumerate</i>	si riferisce ad una lista enumerativa priva di spazi separazione fra una voce e l'altra

(*continua nella pagina successiva*)

Sinossi dei comandi e ambienti specifici della classe (*continua*)

<i>compactdescription</i>	si riferisce ad una lista descrittiva priva di spazi di separazione fra una voce e l'altra; l'etichetta è composta in maiuscoletto
<i>sintassi</i>	Permette di scrivere la sintassi di uno o più comandi dentro un riquadro
<code>guidatematica.bst</code>	file di stile per la bibliografia da gestire con <i>natbib</i> e che ha i riferimenti nello stile autore-anno
<code>guidatematica.ist</code>	file di stile per gli indici analitici con i gruppi di voci che iniziano con la stessa lettera separati dal titolo formato dalla lettera iniziale
<code>guidatematicafiles.ist</code>	analogo ma per un indice analitico tematico relativo solo ai file (vedi pacchetto <i>imakeidx</i>)

Si noti che se non si compone nessun indice analitico i comandi del tipo `\ambstyle` e `\amb`, e le altre coppie analoghe, si comportano nello stesso modo.

I comandi `\showindexmarks` e `\hideindexmarks` talvolta non fanno quello che ci si aspetta; se l'informazione (in corpo `\footnotesize`) è troppo larga per stare completamente nel margine esterno, l'indicazione viene omessa, anche se ce n'è menzione nel file `.log`.

Per maggiori dettagli si vedano i paragrafi specifici che si riferiscono a questi comandi.

La classe si usa nel solito modo dichiarandone il nome nell'argomento del comando `\documentclass`. Non sono preimpostate opzioni di nessun genere e quindi si possono specificare le opzioni che si desiderano fra tutte quelle che la classe *memoir* accetta oltre all'opzione *ipertesto* specifica di questa classe.

Tuttavia bisogna fare alcune semplici considerazioni:

- * Le guide tematiche verosimilmente vengono lette a schermo; quindi è opportuno che siano composte con l'opzione *oneside*; naturalmente non è un obbligo, ma è comodo per il lettore di *non* vedere lo specchio di stampa spostarsi a destra o a sinistra a seconda della parità del numero di pagina.
- * La lettura a schermo è più comoda se il supporto virtuale della pagina, la “carta virtuale”, è di formato B5 invece che A4; si consiglia quindi di comporre la guida tematica con l'opzione *b5paper*, invece dell'opzione *a4paper* o dell'omissione di qualunque opzione relativa al formato della carta.
- * Comunque si ritiene che sia meglio specificare un font normale di 11 punti, perché rende la lettura a schermo più facile.

Per i motivi di immagine esposti nell'introduzione, è opportuno che i singoli utenti apportino il minor numero di personalizzazioni a quanto la classe può produrre da sola.

Naturalmente l'autore è autorizzato ad usare i pacchetti che gli sono funzionali alla composizione della sua specifica guida tematica; viene in mente, per esempio, il pacchetto *tcolorbox* che Roberto Giacomelli ha usato nella sua *Guida tematica alla riga di comando*. Viene in mente il pacchetto *tkzexample* per descrivere le proprietà grafiche del pacchetto *TikZ* o della classe *beamer* su cui Claudio Fiandrino ha scritto la sua

guida tematica *Introduzione alla personalizzazione di Beamer*. Siccome questi pacchetti non sarebbero utili per molte altre guide tematiche, è opportuno che gli autori se li carichino da soli nel preambolo.

Oltre a quanto esposto sopra, si può pensare a qualcosa riguardante i font; per impostazione predefinita vengono usati i font Latin Modern estesi, invocati mediante il pacchetto `cfr-lm` quando si compone con `pdfLaTeX`. Questo pacchetto introduce molti comandi di personalizzazione, tutti usabili anche con questa classe, che permettono, per esempio, di usare i font typewriter proporzionali oppure a spaziatura fissa (predefiniti); di usare le cifre minuscole, oppure maiuscole oppure a spaziatura fissa, specialmente per il corretto incolonnamento nelle tabelle. Ecco, queste sono tutte personalizzazioni “consentite”. Cambiare font, benché tecnicamente possibile, è invece fortemente sconsigliato.

Il layout della pagina e i margini ai quattro lati dello specchio di stampa non sono modificabili; cioè lo sarebbero, ma si raccomanda agli utenti di non farlo; si snaturerebbe l’aspetto grafico della guida, e si romperebbe quell’uniformità grafica richiesta a tutte le guide tematiche del `GIT`.



La classe carica un certo numero di pacchetti di configurazione e di estensione; se ne usino le funzionalità a piacimento, ma non si installino altri pacchetti, talvolta incompatibili, ma tali da alterare l’aspetto grafico della classe.

2.1 PACCHETTI PRECARICATI

Alcuni pacchetti precaricati sono funzionali alla classe stessa, altri sono o potrebbero essere utili per l’utente:

`ifxetex` è funzionale per la classe, visto che si può usare come programma di compilazione sia `pdfLaTeX`, sia `XeLaTeX`. Per esempio, alcuni dei pacchetti che seguono sono funzionali solo se si usa `pdfLaTeX`, quindi essi sono caricati solo se *non* si usa `XeLaTeX`, e viceversa.

`graphicx` serve per inserire e manipolare immagini; disponendo di una distribuzione del sistema `TEX` (TeX Live o MiKTeX molto recente e aggiornata, oltre che completa – non una distribuzione basic, tanto per intenderci) sia `pdfLaTeX` sia `XeLaTeX` possono inglobare lo stesso tipo di file grafici, cioè quelli con estensione `.pdf`, `.jpg`, `.png`, `.eps`, `.mps`. In una prima versione di questa classe si era caricato il pacchetto `guit`, che a sua volta caricava `graphicx`, ma poi si è scoperto

che il disegno del logo  era incompatibile con $\text{X}\mathcal{T}_{\text{E}}\text{X}$; perciò si è preferito *non* caricare questo pacchetto (e l'utente è avvisato di non caricarlo mai, se vuole usare XeLaTeX , finché non sarà aggiornato per essere compatibile con $\text{X}\mathcal{T}_{\text{E}}\text{X}$). Al suo posto si è generata una immagine abbastanza grande del logo, e si è creata una macro che inserisca il logo nelle giuste proporzioni con il corpo del font in uso; cambiando il corpo dei font cambia corrispondentemente anche l'immagine riprodotta del logo .

metalogo serve per poter descrivere i loghi che vengono generalmente usati per indicare certi compilatori o certe indicazioni particolarmente frequenti quando si scrive di $\text{L}\text{A}\text{T}_{\text{E}}\text{X}$. Per esempio consente di usare il comando $\backslash\text{XeLaTeX}$ per scrivere correttamente $\text{X}\mathcal{T}_{\text{E}}\text{X}$ con la 'E' speculare.

pict2e nella versione aggiornata dopo il primo di giugno 2009; è funzionale alla classe, ma potrebbe venire usato anche dall'utente.

microtype serve per usare pienamente la microtipografia se si usa pdfLaTeX con la protrusione e l'espansione dei font; con XeLaTeX bisogna disporre della versione 'beta' di *microtype* che ne consente un uso ridotto ma sufficiente; in particolare con $\text{X}\mathcal{T}_{\text{E}}\text{X}$ non funziona ancora pienamente l'espansione dei font.

etoolbox è funzionale alla classe, ma non è escluso che possa essere utile anche all'utente.

xcolor serve per usare i colori e di sfruttare la sua sintassi per definire i colori "misti"; non è specificata nessuna opzione per i colori. L'utente è pregato di non caricare il pacchetto *color* che vanifica in parte le funzionalità di *xcolor* e può produrre incompatibilità.

natbib è funzionale alla compilazione della bibliografia che usa uno stile "autore-anno" e si rifà ad una variante dello stile bibliografico *arstex-nica.bst* che si chiama *guidatematica.bst*, distribuito insieme a questa classe. Non è vietato usare *biblatex* ma sembra che la cosa sia inutilmente complessa con il tipo di riferimenti bibliografici che si possono citare nelle guide che hanno a che vedere con il mondo di $\text{L}\text{A}\text{T}_{\text{E}}\text{X}$.

multicol può servire all'utente, ma è funzionale alla composizione dell'indice analitico.

imakeidx serve per comporre uno o più indici analitici in modo sincrono con la composizione del documento. Esso era inizialmente incompati-

bile con la classe *memoir* che aveva un suo meccanismo per comporre uno o più indici analitici; dalla versione 1.1 in poi *imakeidx* consente di comporre gli indici analitici anche con *memoir* purché si usino solo le impostazioni e i comandi di *imakeidx*. Dalla versione 1.2e *imakeidx* è compatibile anche con le funzionalità di *showidx* originali, sia con quelle simulate dalla classe *memoir*. Avendo la classe *memoir* come base, non è necessario caricare mai il pacchetto *showidx*, ma per mostrare le voci indicizzate nel margine basta specificare `\showindexmarks` dopo `\begin{document}` per attivare questa funzionalità; `\hideindexmarks` la disattiva; questa funzionalità è inizialmente disattivata ma, appunto, può essere attivata e disattivata dall'utente a suo piacimento.

inputenc è caricato dalla classe con l'opzione di codifica *utf8* se e solo se si sta componendo con pdfLaTeX. Con XeLaTeX si deve usare solo la codifica *utf8* senza specificare il pacchetto *inputenc*. Avendo caricato questo pacchetto con questa opzione, si pensa di avere agevolato l'utente, anche se è possibile che, per comporre con questa classe il file di una guida tematica già scritto in precedenza, egli debba provvedere a cambiarne la codifica.

fontenc è caricato se e solo se si sta componendo con pdfLaTeX; l'opzione prescelta è *TI*; con XeLaTeX si usano solitamente font OpenType in codifica UNICODE, quindi la scelta dei font con cui comporre il documento deve essere fatta dall'utente. Questa classe carica di default i font Latin Modern Type 1 quando si compone con pdfLaTeX e Latin Modern OpenType quando si compone con XeLaTeX.

babel viene caricato solo con l'opzione *italian* e solo se si sta componendo con pdfLaTeX; se si vogliono usare altre lingue oltre all'italiano se ne specifichino le opzioni fra quelle della classe. Se invece si compone con XeLaTeX, il pacchetto *polyglossia* è già precaricato ed è già specificato che la lingua principale è l'italiano, ma si possono dichiarare direttamente altre lingue (secondarie/alternative) secondo le modalità proprie di *polyglossia* con comandi da inserire nel preambolo.

cfr-lm quando si compone con pdfLaTeX serve per usare i font vettoriali Latin Modern in modo avanzato, sfruttandone le molte possibilità di uso che non sono accessibili con il semplice pacchetto *lmodern*. Il lettore si può riferire alla documentazione del pacchetto *cfr-lm* per

avere una panoramica completa di questo uso avanzato che si può fare con questo pacchetto. Quando questa guida viene composta con pdfLaTeX viene usato questo pacchetto *cfr-lm*.

fontspec se invece si compone con XeLaTeX, viene caricato il pacchetto *fontspec* senza opzioni; si sono definiti alcuni font e alcuni comandi per far sì che usare pdfLaTeX oppure XeLaTeX non produca risultati sostanzialmente differenti. Si veda però il capitolo 4.

polyglossia viene richiamato se si compone con XeLaTeX e viene impostato l'italiano come lingua principale; se l'utente desidera usare altre lingue, basta che nel suo preambolo specifichi `\setotherlanguages` elencando le altre lingue che desidera usare con la solita lista di lingue separate da virgole.

fancyvrb può essere utile all'utente per comporre ambienti verbatim in modo diverso da quello che si può ottenere con l'ambiente standard del nucleo di L^AT_EX.

afterpage serve per trattenere l'esecuzione di una sezione di codice fino alla fine della pagina corrente; molto utile in certe circostanze, ma qui finalizzato soltanto all'uso di questa classe.

enumitem serve per personalizzare le liste, crearne di nuove, specificare ogni dettaglio di composizione delle etichette, eccetera; con le funzionalità di questo pacchetto si sono ridefiniti numerosi ambienti utili non solo per comporre questo documento, ma anche per l'utente in generale che usa la classe *guidatematica* per descrivere un tema particolare inerente il mondo di T_EX ed abbia bisogno di nuovi tipi di liste. La consultazione della documentazione di *enumitem* diventa quindi utile e necessaria per usare i comandi di personalizzazione in modo corretto ed efficiente.

listings è il pacchetto principe per comporre listati di comandi di vari programmi; in questa classe esso è già configurato per usare la sintassi valida per il linguaggio T_EX nel suo 'dialetto' L^AT_EX. Possono venire caricati altri dialetti come si possono caricare altri linguaggi. Si possono anche estendere le liste di parole chiave secondo le modalità descritte nella documentazione del pacchetto *listings*.

canoniclayout è funzionale alla classe e definisce il layout della pagina qualunque sia il formato della 'carta'. Il layout della pagina di questa classe è indicato graficamente nella pagina 23.

2.2 NUOVI COMANDI

La classe *guidatematica* mette a disposizione dell'utente un certo numero di comandi per rendere più agevole la descrizione della sintassi di certi comandi e di certi ambienti e, eventualmente, di inviare agli indici analitici i comandi, i nomi dei pacchetti, dei programmi, degli ambienti, eccetera.

`\meta` serve per scrivere il descrittore di un argomento in corsivo fra parentesi acute; per esempio: `\meta{numero}`.

`\marg` serve per racchiudere il descrittore di un argomento fra parentesi graffe composte in carattere typewriter; per esempio: `\marg{numero}`.

`\oarg` serve per racchiudere il descrittore di un argomento fra parentesi quadre composte in carattere typewriter; per esempio: `\oarg{numero}`.

`\garg` richiede due descrittori di argomenti separati da una virgola e serve per comporli fra parentesi tonde, sempre separati da una virgola; virgola e parentesi tonde sono composte in carattere typewriter; per esempio: `\garg{x1,y1}` produce (x_1, y_1) .

`\comando` serve per scrivere un comando verbatim, senza interpretarlo e senza svilupparlo; per esempio: `\textup`.

`\cs` è simile a `\comando` ma richiede il nome del comando senza il backslash iniziale ma ne stampa il nome preceduto dal suo backslash; per esempio: `\normalfont`. Se vengono composti gli indici analitici, invia il comando all'indice analitico generale completo di backslash ma indicizzato senza backslash.

`\Sambiente` serve per descrivere la sintassi di un ambiente, il cui nome ne costituisce il primo argomento; il secondo argomento rappresenta il numero di argomenti che il comando di apertura può ricevere; per esempio:

```
\begin{ambiente}[\langle N-arg \rangle] ... \end{ambiente}
```

Per ambienti che richiedono sia il numero di argomenti, sia il valore di default del primo argomento, se questo è opzionale, bisogna procedere con il prossimo comando.

`\Dambiente` si comporta come il comando precedente ma compone due argomenti facoltativi, di cui il secondo rappresenta il valore di default del primo argomento che si può passare al comando di apertura; per esempio:

```
\begin{<ambiente>}[<N-arg>][<default>] ... \end{<ambiente>}
```

`\Bambiente` serve per comporre in carattere typewriter il comando di apertura di un ambiente specifico; per esempio: `\begin{equation}`.

`\Eambiente` similmente serve per scrivere il comando di chiusura di un ambiente specifico; per esempio: `\end{equation}`.

`\Arg` serve per comporre in carattere typewriter un argomento specifico racchiuso fra le sue parentesi graffe; per esempio: `{Titolo di questo documento}`.

`\GetFileInfo` richiede un argomento costituito dal nome di un file completo di estensione; se il file è dotato del comando `\ProvidesFile`, oppure `\ProvidesClass` oppure `\ProvidesPackage`, allora questo comando estrae le informazioni che attribuisce alle macro `\filename`, `\filedate` e `\fileversion` così da poterle usare per altri scopi, per esempio con il prossimo comando. Il file di cui estrarre le informazioni deve essere già stato caricato; per esempio: il file principale con cui è composto questa guida si chiama `GuidaTematica-doc.tex` e questo nome passato come argomento a `\GetFileInfo` produce le tre informazioni: `\filename = GuidaTematica-doc.tex`; `\filedate = 2012/12/12`; e `\fileversion = v.1.3`. Ma il comando `\documentclass` carica la classe `guidatematica`, perciò il comando `\GetFileInfo{guidatematica.cls}` produce le tre informazioni: `\filename = guidatematica.cls`; `\filedate = 2012/12/12`; e `\fileversion = v.0.6c`.

`\setcopymark` compone il suo argomento ruotato di 90° nella parte bassa del margine esterno; serve per indicare il tipo di bozza che si è appena composta, oppure qualunque altra informazione che si desidera ripetere ad ogni pagina; di per sé il comando non prevede nessun testo predefinito; quindi sta all'utente usare un argomento adatto per avere le informazioni che gli servono; qui, per esempio, dopo aver eseguito il precedente comando `\GetFileInfo` in questa maniera:

```
\GetFileInfo{guidatematica-doc.tex}
```

si usa (solo per questa pagina) il seguente argomento:

```
\setcopymark{\textcolor{blue!50!white}{%
  \filestyle{\filename}, \fileversion, del \filedate}}
```


È ovvio, però, che questo genere di informazioni dovrebbe apparire su tutte le pagine di una bozza e dovrebbe venire cancellato solo per la versione definitiva.

I comandi seguenti definiscono lo stile di scrittura di determinate entità e per scriverle in quello stile; non mettono niente, eventualmente, negli indici analitici. Sarebbe opportuno non ridefinire questi comandi, perché sarebbe desiderabile che in tutte le guide tematiche i nomi delle classi venissero sempre composti in un dato carattere, i nomi dei pacchetti in un altro dato carattere, eccetera, altrimenti i lettori ne resterebbero confusi.

`\ambstyle` serve per comporre il nome di un ambiente in carattere lineare inclinato; per esempio: *description*.

`\classstyle` serve per comporre il nome di una classe in carattere typewriter proporzionale inclinato; per esempio: *guidatematica*.

`\filestyle` serve per comporre il nome di un file in carattere typewriter monospaziato diritto; per esempio: `tlenc.def`.

`\packstyle` serve per comporre il nome di un pacchetto in carattere typewriter proporzionale corsivo; per esempio: *microtype*.

`\progstyle` serve per comporre il nome di un programma in carattere lineare diritto; per esempio: `pdfLaTeX`.

`\opzstyle` serve per comporre il nome di un'opzione in carattere graziato corsivo; per esempio: *landscape*.

`\contastyle` serve per comporre il nome di un contatore in carattere typewriter monospaziato diritto; per esempio: `page`.

`\stylestyle` serve per comporre il nome dello stile di pagina in carattere typewriter monospaziato diritto; per esempio: `headings`.

`\numeristyle` serve per comporre il tipo di numerazione in carattere typewriter monospaziato diritto; per esempio: `arabic`.

`\umisurastyle` serve per comporre il nome di una unità di misura del sistema `TEX` in carattere typewriter monospaziato diritto; per esempio: `pt`.

`\chiavestyle` serve per comporre le varie chiavi che `LATEX` usa nelle scritture del tipo “ $\langle chiave \rangle = \langle valore \rangle$ ” in carattere typewriter monospaziato diritto; per esempio: `angle`.

`\descrittorestyle` serve per comporre i descrittori delle colonne delle tabelle e delle matrici in carattere typewriter monospaziato diritto; per esempio: `p`.

`\posizionestyle` serve per comporre i simboli di posizione degli oggetti flottanti in carattere typewriter monospaziato diritto; per esempio `h`.
`\allineamentostyle` serve per comporre i simboli dell'allineamento verticale di tabelle, matrici, ambienti *minipage*, e simili in carattere typewriter monospaziato diritto; per esempio: `c`.

Va detto che l'uso di questi comandi non è necessario quando non si compongono gli indici analitici per i quali sono funzionali nel senso di comporre la parola nel testo e nell'indice usando lo stesso stile; potrebbe essere necessario usarli nei comandi di sezionamento quando si vuole evitare che i collegamenti ipertestuali generati da *hypertext* (non precaricato nella classe, ma da richiedere espressamente da parte dell'utente mediante l'opzione di classe *iperteso*) non abbiano da “singhiozzare” perché certi comandi non sono accettabili nei testi che possono costituire dei bookmark. In questo caso si dovrebbe procedere usando il comando `\texorpdfstring` in questo modo:

```
\subsection{L'uso della chiave %
             \texorpdfstring{\chiavestyle}{{\angle}}}
```

I comandi seguenti fanno uso dei comandi precedenti e, se è richiesta la composizione degli indici analitici, provvedono a inviare ai debiti file di indicizzazione l'informazione corretta anche quando si tratta di termini da raggruppare sotto una voce principale. Perciò questi comandi sono quasi del tutto equivalenti ai precedenti comandi se *non* è stato specificato il comando `\makeindex` nel preambolo, perché solo questo comando rende attivo ed efficace il comando `\index`.

`\opz` compone il suo argomento nello stile delle opzioni e lo invia all'indice analitico principale sotto la voce “opzione”.

`\conta` compone il suo argomento nello stile dei contatori e lo invia all'indice analitico principale sotto la voce “contatore”.

`\stile` compone il suo argomento come lo stile delle pagine e lo invia all'indice analitico principale sotto la voce “stile della pagina”.

`\numeri` compone il suo argomento nello stile delle numerazioni e lo invia all'indice analitico principale sotto la voce “numerazione”.

`\umisura` compone il suo argomento nello stile delle unità di misura e lo invia all'indice analitico principale sotto la voce “unità di misura”.

- `\chiave` compone il suo argomento nello stile delle chiavi e lo invia all'indice analitico principale sotto la voce “chiave”.
- `\descrittore` compone il suo argomento nello stile dei descrittori di colonna e lo invia all'indice analitico principale sotto la voce “descrittore di colonna”.
- `\posizione` compone il suo argomento nello stile della posizione degli oggetti flottanti e lo invia all'indice analitico principale sotto la voce “Posizione degli oggetti flottanti”.
- `\allineamento` compone il suo argomento nello stile dei codici di allineamento e lo invia all'indice analitico principale sotto la voce “codice di allineamento”.
- `\amb` compone il suo argomento nello stile degli ambienti e lo invia all'indice analitico principale sotto la voce “ambiente”.
- `\class` compone il suo argomento nello stile delle classi e lo invia all'indice analitico principale sotto la voce “classe”.
- `\file` compone il suo argomento nello stile dei file e lo invia all'indice analitico principale sotto la voce “file”.
- `\pack` compone il suo argomento nello stile dei pacchetti e lo invia all'indice analitico principale sotto la voce “pacchetto”.
- `\prog` compone il suo argomento nello stile dei programmi e lo invia all'indice analitico principale sotto la voce “programma”.

2.3 NUOVI AMBIENTI

Grazie al pacchetto *enumitem* si sono fortemente estese le possibilità di personalizzare le liste rispetto a quanto si può fare con i comandi di personalizzazione di *memoir*. I suoi ambienti standard *description* e *blockdescription*, sono stati modificati per avere la loro etichetta composta in maiuscoletto; inoltre sono state definite altre liste che si descrivono qui di seguito.

Si noti che per le liste descrittive si può cambiare a piacere il modo di scrivere l'oggetto da descrivere, e questo torna molto utile in una classe come *guidatematica*, dove vengono descritti spesso oggetti della stessa “categoria”, che spesso debbono essere composti in uno stile particolare oppure devono venire inviati all'indice analitico. Si veda poco oltre alcuni esempi di questi usi che ricorrono alla comoda sintassi del pacchetto *enumitem*, basata su espressioni del tipo $\langle chiave \rangle = \langle valore \rangle$.

DESCRIPTION non differisce dall'omonimo ambiente del nucleo di L^AT_EX e della classe *book*; come è stato ridefinito qui ha l'oggetto da descrivere composto in maiuscoletto.

BLOCKDESCRIPTION è costituito da questa particolare descrizione più compatta orizzontalmente di quella tradizionale di L^AT_EX, ma altrettanto aperta verticalmente, con le etichette non sporgenti a sinistra e senza margini di lista rientrato. Anche in questo ambiente l'oggetto da descrivere è composto in maiuscoletto.

COMPACTITEMIZE produce la solita lista puntata in modo compatto verticalmente, come si è fatto sempre in questa guida. Infatti, tranne in questo particolare ambiente per comporre questa descrizione dei nuovi ambienti, si può notare che gli spazi verticali sono stati eliminati da tutte le liste, e si è lasciato solamente uno spazio di un quarto di riga prima dell'inizio e dopo la fine di tutte le liste.

COMPACTENUMERATE produce una lista numerata verticalmente compatta; differisce da *compactitemize* solo per il modo di scrivere l'etichetta di ogni voce.

COMPACTDESCRIPTION produce una lista descrittiva verticalmente compatta. Si è ritenuto che questo ambiente fosse quello più indicato per descrivere i nuovi comandi introdotti in questa classe e per crearne le corrispondenti voci nell'indice analitico. Lo si è fatto nel modo descritto qui di seguito.

PLAINDESCRIPTION differisce dall'ambiente *description* solo perché il modo di scrivere l'etichetta è impostato a `\normalfont`; questo ambiente si presta particolarmente bene per usare stili diversi per le varie etichette degli oggetti da descrivere.

1. Questo ambiente *plaindescription* o l'ambiente *compactdescription* (come d'altronde si può fare con qualunque altro ambiente gestito con il pacchetto *enumitem*) sono particolarmente adatti per i tipi di ambiente che descrivono comandi o altre cose simili di un linguaggio di programmazione, di una classe, di un pacchetto, perché con il comando di apertura è semplice specificare come argomento facoltativo il modo di comporre l'etichetta dell'oggetto da descrivere; per esempio,

se si volessero descrivere un certo numero di file, varrebbe la pena descriverli tutti con lo stile per comporre i nomi dei file; basta allora fare così:

```
\begin{[plaindescription][format=\ttfamily]
```

usando nella descrizione del formato solo delle dichiarazioni¹; quali dichiarazioni? Quelle usate per definire il comando `\filestyle`. Per la verità la definizione di `\filestyle` contiene una dichiarazione e un comando perché deve venire usata anche nelle specificazioni dell'oggetto da indicizzare, ma non è difficile aprire il file `guidate-matica.cls`, cercare la definizione di `\filestyle` e ricavare da lì le dichiarazioni da usare.

Ci sono sostanzialmente due modi per sfruttare questi ambienti: il primo consiste nell'usare *plaindescription* e scrivere ogni voce della lista nella forma, per esempio:

```
\item[\file{<file>}]
```

Il secondo modo consiste nell'usare l'argomento facoltativo del comando di apertura specificando con la sintassi $\langle \textit{chiave} \rangle = \langle \textit{valore} \rangle$ le opzioni compositive dell'etichetta dell'oggetto da descrivere.

Il vantaggio di seguire questa strada è che dopo aver specificato il $\langle \textit{valore} \rangle$ della chiave `format`, in tutta la lista si userà la normale sintassi `\item[<file>]` per avere tutti i nomi dei file, che iniziano ogni voce della descrizione, composti tutti con lo stesso carattere in modo omogeneo con qualunque altra istanza dello stesso nome di file in qualunque altro punto del documento.

2. In particolare, se si volessero descrivere delle entità particolari da raggruppare nell'indice analitico sotto una stessa voce si potrebbe invocare questo ambiente con il comando:

¹Nella documentazione di *enumitem* c'è scritto che l'ultimo comando che specifica la chiave `format`, o il suo sinonimo `font`, può accettare un argomento; quindi lascerebbe intendere che l'ultima sequenza di controllo potrebbe essere un comando, invece che una dichiarazione. Non spiega però come fare. Si fa così: basta racchiudere fra graffe il contenuto delle quadre che contengono l'oggetto da descrivere. Questo modo di procedere può essere fonte di dimenticanze (bisogna scrivere `\item[{<voce>}]` invece che semplicemente `\item[<voce>]`, quindi è facile dimenticare il paio di graffe interne). Per quanto sia possibile seguire questa strada forse è preferibile seguirne un'altra descritta poco oltre.

```
\begin{compactdescription}%
      [before={\let\makelabel\indicizzafile}]
```

avendo preventivamente definito il comando `\indicizzafile` in questo modo:

```
\DeclareRobustCommand\indicizzafile[1]{\file{#1}}
```

oppure si potrebbe agire direttamente²:

```
\begin{compactdescription}%
      [before={\renewcommand\makelabel[1]{\file{##1}}}]
```

in modo che con l’una o l’altra definizione si potrebbe scrivere:

```
... i file di servizio sono:
\begin{compactdescription}[before={...}]
\item[fleqn.clo] serve per comporre le equazioni ad una
    distanza fissa dal margine sinistro della pagina invece
    che centrate.
\item[leqno.clo] serve per scrivere il numero di ogni
    equazione numerabile a sinistra invece che a destra.
...
```

Naturalmente l’utente, se vuole, può definire altre varianti dell’ambiente di descrizione sfruttando i comandi del pacchetto *enumitem* predisposti a questo scopo.

Sono stati ridefiniti anche gli ambienti *thebibliography* e *theindex*, in modo che compaiano anche nell’indice generale; l’ambiente *theindex* è reso compatibile con il pacchetto *imakeidx* in modo da poterne sfruttare tutte le funzionalità. Probabilmente non era necessario ridefinirlo, ma non disturba l’averlo fatto.

Per la bibliografia si usa lo stile predefinito *guidatematica.bst*, del tipo autore-anno e adattato alla lingua italiana; se si vuole usare *biblatex*

²La chiave **before** mette il suo argomento in testa all’elenco di comandi che L^AT_EX, internamente, esegue per impostare una lista. Il doppio “cancellotto” **##** prima del numero dell’argomento è necessario quando si sta definendo una macro con argomento all’interno di un’altra macro.

o un altro stile, è possibile farlo anche con la presenza di questa impostazione. Per le citazioni l'utente può usare tutte le forme del comando `\cite` e delle sue varianti messe a disposizione dal pacchetto *natbib*. Se l'utente decide di usare *biblatex* troverà comodo specificare l'apposita opzione per rendere *biblatex* compatibile con i suddetti comandi di *natbib*.

Molto importante per una guida di questo genere è l'ambiente *sintassi* predisposto per specificare la sintassi dei comandi dentro un ambiente sbandierato con l'allineamento a sinistra; di per sé questo ambiente definisce l'ambiente *flushleft* e una cornice dentro la quale specificare la sintassi che si desidera descrivere; di solito si tratta di una riga soltanto, ma se dovesse richiedere più righe allora l'utente sarebbe facilitato se in apertura dell'ambiente specificasse il comando `\obeylines` che trasforma ogni fine riga in un "a capo" esplicito. Per esempio si può descrivere l'impostazione di un documento con la sintassi seguente:

```
\documentclass[<opzioni>]{<classe>}
<preambolo>
\begin{document}
<corpo del documento>
\end{document}
```

con i comandi:

```
\begin{sintassi}\obeylines
\cs{documentclass}\oarg{opzioni}\marg{classe}
\meta{preambolo}
\Bambiente{document}
\meta{corpo del documento}
\Eambiente{document}
\end{sintassi}
```

Inoltre la classe *memoir* non definisce l'ambiente *titlepage* per una esplicita scelta del suo autore. Nel nostro caso, invece si vuole che la pagina del titolo sia composta in modo uniforme per tutte le guide tematiche del **GLT**.

Per questo scopo l'utente non deve usare esplicitamente questo ambiente; basta che usi il comando `\maketitle`, che a sua volta imposta l'ambiente. L'utente deve limitarsi a specificare nel preambolo i soliti comandi

arricchiti dall'ulteriore comando `\subtitle`, il cui uso è evidentemente facoltativo:

```
\author{\langle un solo nome di autore o una lista di nomi separati da virgole \rangle}
\title{\langle Titolo della guida \rangle}
\subtitle{\langle eventuale sottotitolo \rangle}
\date{\langle data o altra informazione \rangle}
```

Se `\date` non viene specificato, viene inserita la data del giorno; se viene specificata con un argomento nullo, non viene inserito nulla nella pagina del titolo; altrimenti si può specificare quello che si vuole, purché sia una breve locuzione visto che viene scritta nel piedino. Potrebbe essere utile usare il comando `\GetFileInfo` e inserire in `\date` il nome, la data e la versione contenute nelle rispettive macro.

L'intero blocco dei dati nella pagina del titolo viene centrato orizzontalmente nella pagina nonostante i margini laterali preimpostati per le pagine interne siano diversi. Il logo tondo del `GjT` viene inserito automaticamente; si tratta di una variante del “timbro” del `GjT` predisposta in modo che la scritta attorno al logo centrale non abbia mai le lettere capovolte.

2.4 LA VIRGOLA INTELLIGENTE

Come è noto, i numeri fratti vanno scritti con la virgola nella funzione di separatore decimale in tutte le lingue tranne che in inglese e nelle sue varietà. Certo è possibile usare il pacchetto *siunitx* impostandone le opzioni per usare sempre la virgola e per separare la parte intera e la parte fratta in gruppi di tre cifre collegate da uno spazio fine non separabile (quando le cifre sono più di quattro) usando il suo comando `\num`.

Tuttavia questa classe contiene una definizione della virgola come segno *da usare in matematica* in modo che capisca da solo, dal contesto in cui si trova, se funziona da separatore decimale oppure da segno di interpunzione.

Il fatto che in italiano la virgola matematica svolga due funzioni diverse non è tanto per quel che riguarda i numeri, quanto per le liste di oggetti matematici che compaiono frequentemente quando si scrive tecnico. Queste vanno dall'elenco delle variabili in una funzione (per esempio: $f(x, y, z)$), ad un elenco di funzioni (per esempio: “le funzioni Gx), $H(x)$, e $H(x)$ sono legate da ...”); da una lista di numeri (per esempio: “per ogni $i = 1, 2, 3, \dots$), agli elementi di un insieme (per esempio: $\mathbb{Z}_N = 0, 1, 2, 3, \dots, N$). Gli esempi

forniti non rappresentano una casistica completa, ma sono indicativi del problema.

Anche se la filosofia dell'uso del mark-up di \LaTeX è quella di sollevare il compositore dall'occuparsi della forma, ma di concentrarsi sul contenuto, scrivendo espressioni matematiche è sufficientemente impegnativo occuparsi della moltitudine di simboli diversi, e se si riesce a sollevare il compositore dall'occuparsi della doppia funzione della virgola, tanto di guadagnato.

La macro della virgola intelligente, dunque, usa la virgola decimale se e solo se è seguita da una cifra *senza spazi interposti*, e usa la virgola interpuntiva nelle liste. L'unico caso in cui bisogna stare attenti ad inserire uno spazio dopo la virgola (nel file sorgente) è quando si usa una elencazione di numeri, come per esempio in `\forall i=1,2,3,\dots,\infty` che produce correttamente $\forall i = 1, 2, 3, \dots, \infty$; in questo caso, infatti, il contesto non è abbastanza significativo per l'“intelligenza” della virgola al fine di decidere quale tipo di funzione essa svolga fra due cifre, se non ci fosse lo spazio interposto. Tralasciando gli spazi tra la virgola e la cifra che la segue si otterrebbe invece $\forall i = 1,2,3, \dots, \infty$, che ha le spaziature evidentemente errate. Se ciò che segue la virgola non è una cifra, la definizione della virgola inserisce una virgola interpuntiva, come succede quando essa è seguita dalle sequenze di controllo `\dots` e `\infty`.

Forse è opportuno spiegare perché qui si usi la locuzione “sequenza di controllo” invece della parola “macro” o di altre espressioni più comuni: una sequenza di controllo è in generale un *token complesso*, cioè un token il cui nome è formato da una o più lettere o simboli ed è preceduto dal carattere di escape, generalmente, ma non obbligatoriamente, rappresentato dal segno `\`; potrebbe andare bene qualunque altro carattere che abbia codice di categoria pari a zero. Una sequenza di controllo può essere: (a) indefinita, se non le è mai stato assegnato un significato; (b) una macro se le è stato assegnato un significato mediante il comando primitivo `\def` o un comando equivalente; (c) qualcosa che può avere caratteristiche diverse da quelle di una macro se le è stato attribuito un significato mediante il comando primitivo `\let` o un comando equivalente. In quest'ultimo caso la sequenza di controllo, anche se ha tutto l'aspetto di una macro, può non essere affatto una macro: se per esempio mediante `\let` si rende `\miocomando` equivalente a `\newcommand`, `\miocomando` diventa un sinonimo di `\newcommand` e rappresenta quindi una macro; invece, se me-

dante `\let` si rende `\miaA` equivalente ad `A`, `\miaA` diventa un sinonimo di `A`, quindi rappresenta una lettera, non una macro; infatti in questo caso si dice che la sequenza di controllo è un *carattere implicito*.

Queste sono le situazioni delicate che talvolta rendono difficile la programmazione, perché è facile confondersi, e rendono difficili le spiegazioni, perché la lingua italiana sembra poco flessibile per descrivere questo genere di cose. In relazione a quanto si diceva poco sopra, `\dots` è una macro mentre `\infty` ha le caratteristiche di una macro con pdfL^AT_EX, ma quelle di un carattere implicito con X_YL^AT_EX; perciò la macro della virgola intelligente di questa classe deve saper distinguere le due situazioni sulla base del nome della sequenza di controllo, non solamente sul suo significato.

Non è assolutamente necessario inserire uno o più indici analitici in ogni guida tematica; dipende dall'argomento e dalla complessità di questi indici.

Di solito un solo indice analitico è sufficiente a meno che non si tratti di un indice di parecchie decine di pagine; in questo caso sarebbe forse meglio suddividere l'unico grosso indice analitico in indici parziali suddivisi per argomenti.

Questa classe viene distribuita con due file di stile per l'indice analitico: `guidatematica.ist` e `guidatematicafiles.ist`; entrambi gli stili dividono il loro contenuto in “sezioni” marcate con la lettera iniziale del gruppo di voci. Ma quando l'iniziale non è una lettera, le voci vengono raccolte sotto la definizione di “Simboli”; se l'iniziale fosse una cifra, sarebbe raccolti in una sezione intitolata “Numeri”. Ma se si facesse un indice analitico contenente solo nomi di file e delle loro estensioni (che cominciano con un punto) non sarebbe conveniente raccogliere le estensioni sotto una intestazione “Simboli”, ma è meglio raccoglierle sotto l'intestazione “Estensioni”; questo è quello che fa lo stile definito da `guidatematicafiles.ist`.

Per questo motivo in questa guida l'unico indice analitico è configurato con il seguente comando che usa la sintassi del pacchetto *imakeidx*:

```
\makeindex[options={-s guidatematica.ist},intoc]
```

Come si vede, l'argomento di `options` contiene la specificazione di usare lo stile `guidatematica.ist`, in modo che quando pdfLaTeX o XeLaTeX invia al sistema operativo il comando di eseguire `makeindex` gli passa anche questa opzione.

In questo indice analitico unico i file sono tutti elencati sotto la voce “file” quindi anche le estensioni stanno bene raggruppate sotto questa voce.

La definizione preimpostata nella classe è infatti:

```
\DeclareRobustCommand*\file[1]{\filestyle{#1}%
\index{file!#1@\filestyle{#1}}}
```

Ma se si volesse predisporre un indice dei file usati separato dall'indice analitico generico, allora bisognerebbe specificare:

```
\makeindex[name=files, options={-s guidatematicafiles.ist},
intoc, title=Indice dei file]
```

e i nomi dei file o delle loro estensioni andrebbero specificati con il comando:

```
\index[files]{\langle nome o estensione di un file \rangle}
```

quindi il comando `\file` andrebbe ridefinito così:

```
\DeclareRobustCommand*\file[1]{\filestyle{#1}%
\index[file]{#1@\filestyle{#1}}}
```

Si noti che il comando di definizione `\DeclareRobustCommand` provvede a dichiarare il suo argomento come “robusto”; se il comando è già definito, si limita a mettere un avviso nel file `.log` e non produce un errore come fanno invece i comandi `\newcommand` o `\renewcommand`.

Alcuni utenti considerano molto utile avere l'indicazione delle voci inviate all'indice segnate nel margine della pagina dove la voce da indicizzare è riportata. Può essere utile per togliere i comandi “indicizzanti” dai luoghi dove una certa parola o una certa locuzione è usata, ma non si trova in una posizione importante all'interno del testo; in una posizione, cioè, nella quale si dica qualcosa di importante a proposito della parola o della locuzione. L'utente allora, con l'aiuto di queste indicazioni a margine può facilmente ritrovare le voci indicizzate e analizzare criticamente se quella voce vada riportata nell'indice con riferimento a quella specifica pagina, oppure se non sia importante e possa quindi essere omessa.

Per fare questo esisterebbe il pacchetto *showidx*, ma si raccomanda di non usarlo, perché la classe *memoir* ne provvede le funzionalità, anzi le provvede meglio, nel senso che si possono attivare e disattivare diverse volte lungo il documento. Il comando per attivare questa funzionalità è

`\showindexmarks` e quello per disattivarla è `\hideindexmarks`. Il compositore stia bene attento a disattivare tutte le istanze di attivazione quando compone la versione finale del documento.

Il pacchetto *`imakeidx`* nella versione 1.2e è reso perfettamente compatibile con la funzionalità della classe *`memoir`* e i due comandi di attivazione e disattivazione continuano a funzionare anche quando si usi questo utilissimo pacchetto.

COMPORRE CON pdfLaTeX O CON XeLaTeX 4

Chiaramente la scelta dell'uno o dell'altro programma dipende esclusivamente dall'utente. Vale la pena di segnalare qui alcune differenze per consentire una scelta consapevole.

XeLaTeX differisce da pdfLaTeX, sotto alcuni aspetti sostanziali:

I FONT OPENTYPE I font OpenType sono usabili solo con XeLaTeX; veramente sarebbero usabili anche con pdfLaTeX ma sarebbe necessario creare ed installare opportuni file virtuali per spezzare le numerose pagine di glifi contenute nei file OpenType, in singole pagine di soli 256 caratteri e bisognerebbe ridefinire molti comandi di L^AT_EX per accedere ai singoli frammenti di un font OpenType spezzato come detto sopra.

Disponendo di XeLaTeX (e con le distribuzioni del sistema T_EX successive al 2006 tutte contengono X_TL^AT_EX e i motori di composizione sono disponibili per tutte le piattaforme di elaborazione Windows, Linux, Mac), queste acrobazie con i font si possono evitare facilmente usando, appunto, XeLaTeX.

IL FORMATO DI USCITA Sia pdfLaTeX sia XeLaTeX producono l'uscita in formato PDF; il primo produce quel formato in modo diretto, mentre il secondo passa attraverso una versione estesa e modificata del formato DVI e provvede ad eseguire la conversione in formato PDF mediante il programma esteso xdvpdfmx; siccome il formato DVI, seppure esteso, non consente alcune della funzionalità del formato PDF, la conversione anche se ben fatta, non può introdurre nel file PDF quello che non c'era nel file DVI esteso. Passando, quindi, per XeLaTeX si perdono alcune funzionalità.

IL FORMATO PDF ARCHIVIABILE Non so quanto sia importante per le guide tematiche del G_UI^T, ma con XeLaTeX non si può produrre il

file di uscita in formato PDF compatibile con il formato archiviabile, corrispondente alla norma ISO 19005-1:2005; oggi poi esiste un aggiornamento della norma, ISO 19005-2:2011, e a maggior ragione il file PDF generato con XeLaTeX non è conforme con le norme ISO citate. Tuttavia ricorrendo a **ghostscript** è possibile ottenere la conversione in formato archiviabile in modo corretto; nella sezione di Documentazione del sito del **GNU** esistono alcuni testi che descrivono il modo di procedere. Sembra che con la versione 9.x di **ghostscript** sia possibile generare un file PDF archiviabile conforme alla norma del 2011, ma chi scrive non è ancora in grado di confermarlo.

LA MICROTIPOGRAFIA pdfLaTeX può sfruttare ogni artificio di microtipografia per comporre le pagine in modo praticamente perfetto; con i font in dotazione, può comporre sfruttando sia la protrusione di certi segni completamente o parzialmente nei margini destro e sinistro, sia l'espansione dei font che permette di espandere o comprimere i caratteri di una riga di quel poco che basta per passare inosservato ma che rende gli spazi nella riga molto più omogenei fra righe adiacenti, cosicché non appaiono i brutti “ruscelli” bianchi che serpeggiano tra le parole attraverso la pagina mal composta. XeLaTeX può usare solo la protrusione dei segni, senza l'espansione dei caratteri. Molte delle raffinatezze che il pacchetto *microtype* permette di fare con i caratteri (per esempio spazieggiare il maiuscoletto o il maiuscolo; definire e usare fattori di spazio diversi per i segni di interpunzione, ...) possono essere eseguiti anche con oculate impostazioni delle proprietà dei font OpenType, ma il risultato finale è comunque migliore con l'uso della espansione dei caratteri.

I FONT CON I CORPI OTTICI Il sistema T_EX è nato con i font corredati dei corpi ottici; fin dall'inizio esistevano i font Computer Modern a 5, 6, 7, 8, 9, 10, 12, 14,4, ... punti tipografici. Questo significa che al variare del corpo cambia anche un pochino il disegno dei segni in modo che i più piccoli siano ancora leggibili e i più grandi non appaiano troppo neri. I font Computer Modern sono stati via via sostituiti da font in codifiche diverse ma contenenti più segni (da 128 si è passati a 256); oggi quelli più “moderni” sono i font Latin Modern accessibili normalmente richiamando il pacchetto *lmodern* oppure, ancor meglio, il pacchetto *cfr-lm*; questo consente di selezionare sia font di forma e di serie diverse da quelle tradizionali, sia alcune particolarità non

accessibili con il pacchetto tradizionale. Quasi tutti gli altri font accessibili con pdfLaTeX sono dotati di un unico corpo e tutti gli altri corpi sono ottenuti per ingrandimento o rimpicciolimento proporzionale; il risultato è che spesso i corpi più piccoli sono quasi illeggibili perché i loro tratti sono diventati troppo sottili, e quelli più grandi di serie media ricordano più la serie nera che la serie media, visto che i tratti dei segni sono diventati troppo spessi rispetto allo spazio occupato da ogni singolo segno.

La situazione non è molto diversa con XeLaTeX; la maggior parte dei font OpenType disponibili è dotato di un solo corpo. Fanno eccezione, ovviamente, i font OpenType Latin Modern, che consentono quindi di mantenere la massima leggibilità qualunque sia la dimensione dei caratteri. Se vogliamo, la gestione dei font Latin Modern è leggermente migliore sotto alcuni punti di vista rispetto a quella che si può avere con pdfLaTeX, ma sotto certi altri aspetti richiede un coinvolgimento maggiore da parte dell'utente. Esistono anche altri pochi font dotati di corpi ottici, e spesso il loro uso non è completamente libero come avviene per i Latin Modern.

Infatti per avere le proprietà dei font Latin Modern in formato Type 1 usabili da pdfLaTeX bastano poche righe di codice per caricarli e usarli:

- * nel preambolo basta usare un solo statement con le caratteristiche desiderate; qui si è usato:

```
\RequirePackage[tt={oldstyle=false,tabular,monowidth}]{cfr-lm}
```

Con questa unica riga si dispone dell'intera collezione con le impostazioni di default, tranne che per il font typewriter per il quale non si vogliono le cifre minuscole (old style), ma si vogliono quelle maiuscole adatte alle tabelle, e si vuole la varietà di font monospaziati.

- * in questa classe sono anche già disponibili i comandi `\textttv` per selezionare il font typewriter proporzionale e `\textttm` per usare quello a spaziatura fissa indipendentemente dalle impostazioni specificate fra le opzioni del pacchetto; inoltre il comando `\textl` garantisce l'uso delle cifre maiuscole con qualunque famiglia, forma e serie.

Per avere le stesse funzionalità con XeLaTeX si sono dovute specifi-

care le famiglie dei font: principale (font proporzionale con grazie e le cifre minuscole); sans serif (font proporzionale sgraziato o lineare, che dir si voglia, con le cifre minuscole); typewriter proporzionale con le cifre maiuscole; typewriter a spaziatura fissa con le cifre maiuscole; infine il maiuscoletto con le cifre minuscole.

Con queste impostazioni si è però ridefinito il comando `\textl` per le cifre maiuscole tenendo conto delle particolarità dei font OpenType, e con `\DeclareTextFontCommand` si sono dovuti definire i comandi equivalenti a quelli disponibili con il pacchetto *cfr-lm* usabile solo da pdfLaTeX.

```
\RequirePackage{fontspec}
\setmainfont[Ligatures=TeX,Numbers=OldStyle]%
                                     {Latin Modern Roman}
\setsansfont[Ligatures=TeX,Numbers=OldStyle]{Latin Modern Sans}
\setmonofont[Numbers=Lining]{Latin Modern Mono}
\newfontfamily{\tvfamily}[Numbers=Lining]%
                                     {Latin Modern Mono Prop}
\newfontfamily{\scfamily}[Ligatures=TeX,Numbers=OldStyle]%
                                     {Latin Modern Roman Caps}
\DeclareRobustCommand\scshape{\scfamily}
\DeclareTextFontCommand{\texttm}{\ttfamily}
\DeclareTextFontCommand{\texttv}{\tvfamily}
\DeclareTextFontCommand{\textsc}{\scfamily}
\def\textl#1{{\addfontfeature{Numbers=Lining}#1}}
```

In compenso è stato possibile specificare gli stessi font Latin Modern adatti alla matematica, in modo da sfruttare appieno le potenzialità dei font OpenType matematici; il pacchetto *unicode-math* mette a disposizione tutti i comandi matematici necessari.

```
\RequirePackage{unicode-math}
\setmathfont{Latin Modern Math}
```

In questo modo comporre la documentazione con questa classe produce sostanzialmente gli stessi risultati che si usi pdfLaTeX oppure XeLaTeX.

Perché scegliere l'uno o l'altro programma di composizione? Dipende dal contenuto della guida tematica; se occorre fare uso di alfabeti diversi da quello latino (per esempio: greco, cirillico, arabo, eccetera) allora XeLaTeX è obbligatorio. Altrimenti è preferibile usare pdfLaTeX.

BIBLIOGRAFIA

Per concludere si riporta qui una bibliografia che si suppone sia utile per la documentazione non tanto di questa classe, quanto per chi scrive guide tematiche sui vari argomenti che possono interessare i frequentatori del sito del [GJIT](#).

* * *

L^AT_EX 3 PROJECT TEAM (2006). «L^AT_EX 2_ε for class and package writers». PDF document. Leggibile con `texdoc clsguide` nella distribuzione TeX Live.

BECCARI, C. e GREGORIO, E. (2010). «Package *imakeidx*». PDF document. Leggibile con `texdoc imakeidx` nella distribuzione TeX Live.

BERRY, K. (2011). «The T_EX Live guide». PDF document. Leggibile con `texdoc texlive` nella distribuzione TeX Live.

BEZOS, J. (2011). «Customizing lists with the *enumitem* package». PDF document. Leggibile con `texdoc enumitem` nella distribuzione TeX Live.

BRINGHURST, R. (2004). *The elements of typographic style*. Hartley & Marks.

CARLISLE, D. (2007). «The file `ltxdoc.dtx` for use with L^AT_EX 2_ε». PDF document. Leggibile con `texdoc ltxdoc` nella distribuzione TeX Live.

CHARETTE, F. (2010). «Polyglossia: a babel replacement for X_YL^AT_EX». PDF document. Leggibile con `texdoc polyglossia` nella distribuzione TeX Live.

- GÄSSLEIN, H., NIEPRASCHK, R. e TKADLEC, J. (2011). «The *pict2e* package». PDF document. Leggibile con `texdoc pict2e` nella distribuzione TeX Live.
- GREGORIO, E. (2009). «Appunti di programmazione in \LaTeX e \TeX ». PDF document. URL <http://profs.sci.univr.it/~gregorio/egtex.html>.
- KNUTH, D. E. (1984). *The \TeX book*. Addison Wesley. Nonostante l'anno di pubblicazione nell'arco degli anni sono state stampate molte ristampe rivedute e corrette.
- KOPKA, H. e DALY, P. W. (2004). *Guide to \LaTeX* . Addison Wesley, 4^a edizione.
- LAMPORT, L. (1994). *\LaTeX : A document preparation system – User guide and reference manual*. Addison Wesley, 2^a edizione.
- LEHMAN, P. (2011a). «The *biblatex* package». PDF document. Leggibile con `texdoc biblatex` nella distribuzione TeX Live.
- (2011b). «The *etoolbox* package». PDF document. Leggibile con `texdoc etoolbox` nella distribuzione TeX Live.
- MITTELBACH, F. *et al.* (2003). «The DocStrip program». PDF document. Leggibile con `texdoc docstrip` nella distribuzione TeX Live.
- PAKIN, S. (2004). «How to package your \LaTeX package». PDF document. Leggibile con `texdoc dtxut` nella distribuzione TeX Live.
- UMEKI, H. (2010). «The *geometry* package». PDF document. Leggibile con `texdoc geometry` nella distribuzione TeX Live.
- WILSON, P. (2009). «A few notes on book design». PDF document. Leggibile con `texdoc memdesign` nella distribuzione TeX Live.
- WILSON, P. e MADSEN, L. (2011). «The *memoir* class for configurable typesetting – User guide». PDF document. Leggibile con `texdoc memoir` nella distribuzione TeX Live.

INDICE ANALITICO

A

`\allineamento`, 15, 27
`\allineamentostyle`, 14, 15, 26
`\amb`, 15, 17, 27
ambiente
 array, 14
 blockdescription, 16
 compactdescription, 28
 compactdescription, 17
 compactenumerate, 16
 compactitemize, 16, 28
 description, 16
 flushleft, 31
 minipage, 8, 26
 plaindescription, 16, 28, 29
 sintassi, 17, 31
 tabular, 14
 thebibliography, 30
 theindex, 30
 titlepage, 31
`\ambstyle`, 13, 15, 17, 25
`\Arg`, 12, 24
`\author`, 6, 9, 32

B

`\backmatter`, 6, 9
`\Bambiente`, 11, 24
`\begin`, 6, 11, 21, 23, 24, 31
`\bibliography`, 7

C

`\chapter`, 6
chiave
 before, 30
 font, 29
 format, 29
`\chiave`, 15, 27
`\chiavestyle`, 14, 15, 25
`\cite`, 31
`\class`, 16, 27
classe
 beamer, 18
 book, 28
 guidatematica, 2, 3, 9, 22–24, 27
 memoir, 2, 18, 21, 27, 31, 36, 37
`\classstyle`, 13, 16, 25
`\comando`, 11, 23
`\conta`, 14, 26

`\contastyle`, 14, 25

`\cs`, 11, 23

D

`\Dambiente`, 11, 23

`\date`, 8, 9, 32

`\DeclareRobustCommand`, 36

`\DeclareTextFontCommand`, 41

`\def`, 33

`\descrittore`, 15, 27

`\descrittorestyle`, 14, 15, 25

`\documentclass`, 31

`\documentclass`, 18, 24

`\dots`, 34

E

`\Eambiente`, 11, 24

`\end`, 7, 11, 23, 24, 31

F

file

`.dtx`, 2

`.eps`, 19

`.jpg`, 19

`.log`, 17, 36

`.mps`, 19

`.pdf`, 19

`.png`, 19

`.tex`, 7

`arstexnica.bst`, 20

`GuidaTematica-doc.pdf`, 5

`GuidaTematica-doc.tex`, 24

`guidatematica.bst`, 17, 20, 30

`guidatematica.cls`, 29

`guidatematica.dtx`, 5

`guidatematica.ist`, 17, 35

`guidatematica.pdf`, 5

`guidatematica.zip`, 5

`guidatematicafiles.ist`, 17,

35

`\file`, 16, 27, 29, 36

`\filedate`, 9, 12, 24

`\filename`, 9, 12, 24

`\filestyle`, 13, 16, 25, 29

`\fileversion`, 9, 12, 24

`\frontmatter`, 6, 9

G

`\garg`, 11, 23

`\GetFileInfo`, 9, 12, 24, 32

H

`\hideindexmarks`, 12, 17, 21, 37

I

`\index`, 26, 36

`\indicizzafile`, 30

`\infty`, 34

`\item`, 29

L

`\let`, 33, 34

M

`\mainmatter`, 6, 9
`\makeindex`, 6, 12, 26
`\maketitle`, 6, 8, 9, 12, 31
`\marg`, 11, 23
`\meta`, 11, 23
`\misura`, 15

N

`\newcommand`, 36
`\newpage`, 6
`\normalfont`, 23, 28
`\num`, 32
`\numeri`, 15, 26
`\numeristyle`, 14, 25

O

`\oarg`, 11, 23
`\obeylines`, 31
`\opz`, 14, 26
 opzione
 `a4paper`, 18
 `b5paper`, 18
 `iperteso`, 26
 `ipertesto`, 8, 18
 `italian`, 21
 `oneside`, 18
 `T1`, 21
 `utf8`, 21
`\opzstyle`, 13, 14, 25

P

pacchetto
 `afterpage`, 22
 `babel`, 21
 `biblatex`, 20, 30, 31
 `canoniclayout`, 22
 `cfr-lm`, 19, 21, 22, 39, 41
 `color`, 20
 `enumitem`, 2, 22, 27–30
 `etoolbox`, 20
 `fancyvrb`, 22
 `fontenc`, 21
 `fontspec`, 22
 `graphics`, 19
 `guit`, 19
 `hyperref`, 8
 `hypertext`, 26
 `ifxetex`, 19
 `imakeidx`, 10, 12, 17, 20, 21, 30, 35, 37
 `inputenc`, 21
 `listings`, 22
 `lmodern`, 21, 39
 `metalogo`, 20
 `microtype`, 20, 39
 `multicol`, 20
 `natbib`, 17, 20, 31
 `pict2e`, 20
 `polyglossia`, 21, 22
 `showidx`, 21, 36
 `siunitx`, 32
 `tcolorbox`, 18
 `TikZ`, 18
 `tkzexample`, 18
 `unicode-math`, 41
 `xcolor`, 20

- `\pack`, 16, 27
- `\packstyle`, 13, 16, 25
- `\posizione`, 15, 27
- `\posizionestyle`, 14, 15, 26
- `\prebibhook`, 7, 9, 13
- `\printindex`, 7
- `\prog`, 16, 27
- programma
 - ghostscript, 39
 - makeindex, 35
 - pdfLaTeX, 4, 7, 19–22, 35, 38–41
 - xdvpdfmx, 38
 - XeLaTeX, 4, 7, 19–22, 35, 38–41
- `\progstyle`, 13, 16, 25
- `\ProvidesClass`, 24
- `\ProvidesFile`, 6, 8, 24
- `\ProvidesPackage`, 24
-
- R
 - `\renewcommand`, 9, 36
-
- S
 - `\Sambiente`, 11, 23
 - `\setcopymark`, 12, 24
-
- `\setotherlanguages`, 22
- `\showindexmarks`, 12, 17, 21, 37
- `\stile`, 15, 26
- `\stilestyle`, 14, 15, 25
- `\subtitle`, 9, 12, 32
-
- T
 - `\tableofcontents`, 6
 - `\texorpdfstring`, 26
 - `\textl`, 13, 40, 41
 - `\texttm`, 12, 40
 - `\textttv`, 13
 - `\textttv`, 13, 40
 - `\textwidth`, 8
 - `\title`, 6, 9, 32
 - `\tvfamily`, 13
-
- U
 - `\umisura`, 26
 - `\umisurastyle`, 14, 25
-
- X
 - `\XeLaTeX`, 20