Una (mica tanto) breve introduzione a $\LaTeX\,2_\varepsilon$

Ovvero $\LaTeX\,2_\varepsilon$ in 84 minuti

Tobias Oetiker

Hubert Partl, Irene Hyna e Elisabeth Schlegl

Versione 3.16, 25 settembre 2000

Titolo originale: *The Not So Short Introduction to $\LaTeX 2_\varepsilon$*

Traduzione:

Giulio Agostini, Giuseppe Bilotta, Flavio Casadei Della Chiesa,
Onofrio de Bari, Giacomo Delre, Luca Ferrante, Tommaso Pecorella,
Mileto Rigido, Roberto Zanasi.

Ringraziamenti

Molto del materiale utilizzato in questo documento proviene da un'introduzione austriaca a LaTeX 2.09 scritta in tedesco da:

> Hubert Partl `<partl@mail.boku.ac.at>`
> *Zentraler Informatikdienst der Universität für Bodenkultur Wien*
>
> Irene Hyna `<Irene.Hyna@bmwf.ac.at>`
> *Bundesministerium für Wissenschaft und Forschung Wien*
>
> Elisabeth Schlegl `<no email>`
> *in Graz*

Se siete interessati al documento tedesco, potete trovarne una versione aggiornata per il LaTeX 2_{ε} da Jörg Knappen presso `CTAN:/tex-archive/info/lshort/german`

Nel corso della stesura di questo documento ho cercato dei revisori su `comp.text.tex` ricevendo numerose risposte. Le seguenti persone mi hanno aiutato, con grande impegno, a dare a questo documento la sua forma attuale, inviandomi correzioni, suggerimenti e materiale per migliorarlo. Vorrei ringraziare sinceramente ciascuno di loro. Naturalmente tutti gli errori che troverete in questo libro sono miei; se invece doveste trovare anche soltanto una parola scritta correttamente... deve essere stata una delle persone qui sotto che mi ha passato due righe.

Rosemary Bailey, Friedemann Brauer, David Carlisle,
Mike Chapman, Christopher Chin, Chris McCormack,
Wim van Dam, Jan Dittberner, Michael John Downes,
David Dureisseix, Elliot, David Frey, Robin Fairbairns, Erik Frisk,
Frank, Alexandre Guimond, Cyril Goutte, Greg Gamble,
Neil Hammond, Rasmus Borup Hansen, Markus Brühwiler, Björn
Hvittfeldt, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni,
Alan Jeffrey, Byron Jones, David Jones, Johannes-Maria Kaltenbach,
Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen,
Kjetil Kjernsmo, Maik Lehradt, Alexander Mai, Martin Maechler,
Claus Malten, Kevin Van Maren, Lenimar Nunes de Andrade,
Hubert Partl, John Refling, Mike Ressler, Brian Ripley,
Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Hanspeter Schmid,
Craig Schlenter, Christopher Sawtell, Josef Tkadlec, Didier Verna,
Fabian Wernli, Carl-Gustav Werner, Chris York, Fritz Zaucker,
Rick Zaccone, e Mikhail Zotov.

Prefazione

LaTeX [1] è un sistema di composizione di testo particolarmente indicato per la produzione di documenti scientifici e matematici di elevata qualità tipografica, ma è anche adatto a produrre tutti gli altri tipi di documenti, dalle semplici lettere ai libri più completi. LaTeX usa TeX [2] come motore di formattazione.

Questa breve introduzione descrive LaTeX 2_ε e dovrebbe essere sufficiente per la maggior parte degli utilizzi di LaTeX. Per una descrizione completa del sistema LaTeX si può fare riferimento a [1, 3].

LaTeX è disponibile per la maggior parte dei computer, dai PC ai Mac, fino ai grandi sistemi UNIX e VMS.

In molti centri di calcolo universitari si può trovare una installazione di LaTeX pronta all'uso; informazioni su come accedere alla installazione locale di LaTeX dovrebbero essere fornite nella cosiddetta *Local Guide* [4]; se avete difficoltà a cominciare, chiedete alla persona che vi ha dato questo libretto. Lo scopo di questo documento *non* è spiegare come installare e configurare un sistema LaTeX, ma insegnare come scrivere documenti in modo che possano essere elaborati da LaTeX.
Questa Introduzione è divisa in cinque capitoli:

Il **Capitolo 1** descrive la struttura di base dei documenti LaTeX 2_ε. Si apprenderà anche qualche elemento della storia di LaTeX. Dopo aver letto questo capitolo, si dovrebbe avere un'idea approssimativa di LaTeX; sarà solo una base di partenza, ma permetterà di integrare le informazioni fornite negli altri capitoli in un quadro più grande.

Il **Capitolo 2** entra nei dettagli della composizione dei documenti spiegando la maggior parte dei comandi e degli ambienti di LaTeX. Dopo aver letto questo capitolo, si sarà in grado di scrivere i primi documenti.

Il **Capitolo 3** spiega come comporre formule con LaTeX. Anche qui, molti esempi aiutano a capire come usare uno dei maggiori punti di forza di LaTeX. Alla fine di questo capitolo si trovano delle tavole che elencano i simboli matematici disponibili in LaTeX.

Il **Capitolo 4** spiega la creazione dell'indice e della bibliografia, l'inclusione di grafica EPS e alcune altre utili estensioni.

Il **Capitolo 5** contiene alcune informazioni potenzialmente pericolose su come modificare il formato predefinito dei documenti prodotti da LaTeX. Spiegherà come cambiare le impostazioni in modo che, volendo, i buoni risultati che si possono ottenere con LaTeX diventino decisamente... più brutti!

È importante leggere i capitoli uno ad uno nel loro ordine: dopo tutto il libro non è così lungo come sembra. Ci si assicuri di leggere con cura gli esempi proposti nel corso del libro, perché in essi è contenuta una gran parte delle informazioni.

Se cercate del materiale riguardante LaTeX, date uno sguardo a un archivio ftp del Comprehensive TeX Archive Network (`CTAN`). Il materiale può essere trovato, per esempio, su `ctan.tug.org` (Stati Uniti), `ftp.dante.de` (Germania), `ftp.tex.ac.uk` (Regno Unito). Se non vi trovate in uno di questi Paesi, scegliete l'archivio più vicino.[1]

Se volete utilizzare LaTeX sul vostro computer, date uno sguardo a quanto è disponibile presso `CTAN:/tex-archive/systems`.

Se avete idee su cosa aggiungere, togliere o modificare in questo documento, per favore fatemelo sapere. Mi interessano specialmente i commenti dei principianti di LaTeX su quali parti di questa Introduzione risultino di facile comprensione e quali invece potrebbero essere spiegate meglio.

Tobias Oetiker `<oetiker@ee.ethz.ch>`

Department of Electrical Engineering,
Swiss Federal Institute of Technology

La versione in inglese più aggiornata di questo documento è disponibile presso `CTAN:/tex-archive/info/lshort`

[1]Per l'Italia: `ftp.unina.it` e `cis.uniRoma2.it` [N.d.T.]

Nota dei traduttori

La traduzione di questo documento è stata portata a termine da alcuni frequentatori del *newsgroup* `it.comp.software.tex`, che hanno dato al loro gruppo il nome "Gilda Guild" (letto come sigla, diventa *Gruppo Italiano LaTeX: Documentazione e Altro / General pUrpose Italian LaTeX Domain*). In questa breve nota essi desiderano far conoscere al lettore i motivi che hanno portato ad alcune scelte stilistiche e un consiglio per chi utilizza la tastiera italiana.

Spesso, nei testi di lingua italiana, ci si riferisce alle diverse parti di un capitolo con il termine "paragrafo", e raramente "sottoparagrafo". In questa traduzione, tuttavia, si è optato per traduzioni dei termini relativi ai comandi di sezionamento più aderenti all'originale, riferendosi quindi ad una `\section` con il termine *sezione*, ad una `\subsection` con il termine *sottosezione*, ad un `\paragraph` con il termine *paragrafo*, e ad una parte di testo delimitata da due righe vuote o da comandi `\par` con il termine *capoverso*. Con ciò si spera di non aver generato ulteriore confusione riguardo a questo punto spinoso.

I termini "*superscript*" e "*subscript*" sono stati tradotti con *apice* e *pedice*, o con *esponente* e *deponente*, privilegiando la seconda traduzione in ambito matematico.

In altri casi, sono state preferiti termini di uso comune (intestazione, piè di pagina) a termini forse più corretti, ma raramente adottati in questo ambito (testatina, riga in calce).

Chi scrive con una tastiera italiana, può avere problemi a digitare i caratteri `{`~}`, che non compaiono su alcun tasto. Può giovare sapere che su alcuni sistemi operativi, questi caratteri possono essere ottenuti tenendo premuto il tasto ALT sinistro e digitando il codice ASCII relativo con il tastierino numerico. Per comodità si riportano i codici relativi a questi quattro caratteri. In alcuni casi, le parentesi graffe si ottengono premendo

Carattere	`	{	}	~
Codice	96	123	125	126

Tabella 1: Codici ASCII relativi ad alcuni caratteri frequenti.

ALT GR+MAIUSC+'è' e ALT GR+MAIUSC+'+' (ovvero come se si dovessero fare delle parentesi quadre "maiuscole").

Si desidera ringraziare Marco d'Itri e Andrea Brugiolo per avere contribuito alla traduzione di questa guida mettendo a disposizione un server CVS e una *mailing-list*.

Le traduzioni del gruppo Gilda Guild si trovano ai seguenti indirizzi:

- `http://guild.prato.linux.it`
- `ftp://lorien.prato.linux.it/pub/guild`
- `ftp://ftp.unina.it/pub/TeX/info/italian`

e su ogni sito CTAN, per esempio in `ftp://ftp.tex.ac.uk/tex-archive/info/italian`.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Cosa è necessario sapere

Nella prima parte di questo capitolo sarà offerta una breve visione d'insieme relativamente alla filosofia e alla storia del LaTeX 2_ε. La seconda parte del capitolo si sofferma invece sulle strutture basilari di un documento LaTeX. Dopo aver letto questo capitolo, si dovrebbe avere una conoscenza di base sul funzionamento del LaTeX; procedendo nella lettura, questo aiuterà a integrare tutte le nuove informazioni in una visione generale d'insieme.

1.1 Il nome del gioco

1.1.1 TeX

TeX è un programma realizzato da Donald E. Knuth [2], ed è concepito per la composizione di testo e formule matematiche. Knuth iniziò a scrivere il motore di tipocomposizione TeX nel 1977 allo scopo di esplorare le potenzalità dell'equipaggiamento digitale di stampa che stava iniziando a prendere piede nel campo dell'editoria a quel tempo, in particolare con la speranza di poter far regredire la tendenza al deterioramento della qualità tipografica che constatava affliggere i suoi libri ed articoli. Il TeX per come è usato oggi è stato pubblicato nel 1982, con alcuni piccoli miglioramenti aggiunti nel 1989 per perfezionare il supporto dei caratteri a 8 bit e di più lingue. Il TeX è rinomato per essere estremamente stabile, eseguibile su diversi tipi di computer e virtualmente privo di bug. Il numero di versione del TeX converge a π ed attualmente è 3.14159.

TeX si pronuncia "Tech", con un "ch" come nella parola tedesca "Ach" o in quella scozzese "Loch". In un ambiente ASCII, TeX diventa `TeX`.

1.1.2 LaTeX

Il LaTeX è un pacchetto di macro che permette agli autori di impaginare e stampare il loro lavoro ai più elevati livelli di qualità tipografica, servendosi

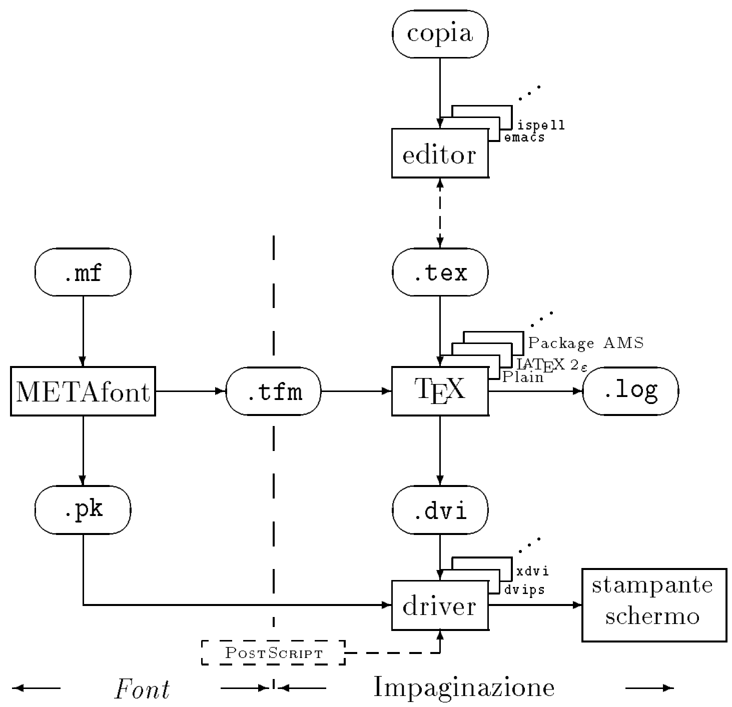


Figura 1.1: Componenti di un sistema TEX

di un layout professionale predefinito. Il LaTeX fu originariamente scritto da Leslie Lamport [1]; esso si serve del TEX come motore di tipocomposizione.

Nel 1994 il pacchetto LaTeX è stato aggiornato dal team LaTeX3, guidato da Frank Mittelbach, al fine di includere alcuni miglioramenti da lungo tempo richiesti e allo scopo di riunire tutte le versioni con *patch* che si erano raccolte sin dalla pubblicazione di LaTeX 2.09 qualche anno prima. Per distinguere la nuova versione da quella antica, essa è chiamata LaTeX 2_ε; questa documentazione tratta il LaTeX 2_ε.

LaTeX si pronuncia "Latech[1]". Se si fa riferimento a LaTeX in un ambiente `ASCII`, si scriverà `LaTeX`. LaTeX 2_ε si pronuncia "Latech due e" e si scrive `LaTeX2e`.

La figura 1.1 in alto mostra come TEX e LaTeX 2_ε operano congiuntamente. Questa figura è tratta da `wots.tex` di Kees van der Laan.

[1] Si tratta della tipica pronuncia italiana. [N.d.T.]

1.2 Concetti basilari

1.2.1 L'autore, il grafico editoriale e il tipografo

Per pubblicare qualcosa, gli autori consegnano il loro manoscritto dattilografato a una casa editrice. Uno dei grafici editoriali stabilisce quindi il layout del documento (la larghezza delle colonne, i *font*, lo spazio prima e dopo le intestazioni,...). Il grafico editoriale riporta quindi le proprie istruzioni nel manoscritto e lo consegna al tipografo, che impagina il libro in base a queste istruzioni.

Un grafico editoriale in carne e ossa cerca di scoprire cosa aveva in mente l'autore durante la realizzazione del manoscritto; egli stabilisce i titoli dei capitoli, le citazioni, gli esempi, le formule, etc. basandosi sulla sua competenza professionale e sui contenuti del manoscritto.

In un ambiente LaTeX, LaTeX impersona il ruolo del grafico editoriale e si serve di TeX come tipografo, ma LaTeX è "solo" un programma e necessita di conseguenza di maggiore assistenza; l'autore deve quindi fornire informazioni aggiuntive che descrivono la struttura logica del documento. Queste informazioni sono inserite nel testo come "comandi LaTeX".

Questo modo di agire è piuttosto diverso dall'approccio WYSIWYG[2] che hanno la maggior parte degli attuali *word processor* come *MS Word* o *Corel WordPerfect*; con queste applicazioni, gli autori specificano il layout del documento interattivamente durante la digitazione del testo al computer e, strada facendo, possono osservare sullo schermo come apparirà l'opera finita quando sarà stampata.

Quando si usa LaTeX, non è normalmente possibile osservare l'output definitivo mentre si digita il testo, ma esso può essere visualizzato preventivamente sullo schermo dopo aver elaborato il file con LaTeX; le correzioni saranno poi effettuate prima di inviare materialmente il documento alla stampante.

1.2.2 Il progetto del layout

Il design tipografico richiede abilità. Gli autori inesperti commettono spesso gravi errori di formattazione poiché ritengono che l'impostazione di un libro sia per la maggior parte una questione di estetica—"Se un documento appare bello esteticamente, allora è ben progettato". Poiché però un documento deve esser letto e non esposto in una pinacoteca, la leggibilità e la comprensibilità sono di maggiore importanza rispetto al suo aspetto gradevole. Esempi:

- La dimensione dei *font* e la numerazione dei titoli sono state scelte per rendere chiara al lettore la struttura dei capitoli e delle sezioni.

[2]"What You See Is What You Get", ciò che vedi è ciò che ottieni.

- La lunghezza delle righe deve essere abbastanza ridotta da non affaticare gli occhi di chi legge, e al contempo sufficientemente abbondante in modo da riempire la pagina con un bell'aspetto.

Con i sistemi WYSIWYG, gli autori creano frequentemente documenti gradevoli sotto il profilo estetico ma con una struttura molto ridotta o incoerente. Il LaTeX mette al riparo da tali errori di formattazione obbligando l'autore a dichiarare la struttura *logica* del proprio documento, scegliendo successivamente il layout più appropriato.

1.2.3 Vantaggi e svantaggi

Quando la gente del mondo WYSIWYG incontra gli utenti di LaTeX, si finisce a discutere spesso su "i vantaggi di LaTeX nei confronti di un normale *word processor*" o viceversa. La migliore cosa da fare quando una tale diatriba ha inizio è mantenere un'impostazione misurata, poiché tali discussioni sfuggono spesso di mano, ma talvolta non ci si può sottrarre...

Ecco qui alcuni argomenti. I principali vantaggi di LaTeX sui normali *word processor* sono i seguenti:

- Sono reperibili dei layout realizzati in maniera professionale, che fanno realmente apparire un documento come se fosse stato "stampato".
- La composizione tipografica di formule matematiche è supportata in maniera comoda.
- L'utente deve solo imparare pochi comandi facili da comprendere che specificano la struttura logica di un documento; egli non avrà mai bisogno di aggiustare alla meglio l'effettivo layout del documento.
- Persino strutture complesse come le note a piè di pagina, i riferimenti, gli indici e le bibliografie possono essere generate con facilità.
- Esistono diversi pacchetti aggiuntivi gratuiti per molti compiti tipografici non direttamente supportati dal LaTeX. Sono reperibili ad esempio dei pacchetti per includere grafici PostScript o per comporre bibliografie conformi a precisi standard. Molti di questi pacchetti aggiuntivi sono descritti in *The LaTeX Companion* [3].
- Il LaTeX incoraggia gli autori a scrivere documenti ben strutturati, poiché è questo il modo in cui LaTeX opera—specificando la struttura.
- Il TeX, il motore di tipocomposizione di LaTeX 2_ε, è gratuito e altamente portabile; il sistema, di conseguenza, è eseguibile su quasi tutte le piattaforme hardware disponibili.

Il LATEX ha anche alcuni svantaggi, e credo che sia difficile per me trovarne di rilevanti, sebbene sia sicuro che altre persone potrebbero elencarne a centinaia ;-)

- Il LATEX non lavora bene per la gente che ha venduto la propria anima...
- Sebbene alcuni parametri possano essere ritoccati nell'ambito del layout predefinito di un documento, l'impostazione di un layout interamente nuovo è complicata e richiede molto tempo[3].
- È molto difficile scrivere documenti non strutturati e disomogenei.
- Il tuo criceto potrebbe, a dispetto di alcuni incoraggianti primi passi, non essere mai capace di comprendere appieno il concetto di "logical markup".

1.3 I file di input di LATEX

L'input per LATEX è un file di testo in `ASCII` puro. Può essere creato con qualunque *editor* di testi e contiene il testo del documento così come i comandi che comunicano al LATEX come impostare il testo.

1.3.1 Spazi

I caratteri di "spazio interparola" come lo spazio vuoto o il carattere di tabulazione sono trattati indifferentemente come "spazio" da LATEX. *Più caratteri consecutivi* di spazio interparola sono considerati come *uno* "spazio". Lo spazio interparola all'inizio di una riga viene generalmente ignorato, e una sola interruzione di riga è trattata come uno "spazio interparola".

Una riga vuota tra due righe di testo delimita la fine di un paragrafo. *Più* righe vuote sono considerate alla stessa maniera di *una* riga vuota. Il testo in basso è un esempio. Nella parte sinistra vi è il testo del file di input, e a destra è visualizzato l'output formattato.

```
Non ha importanza se
immetti uno o tanti     spazi
dopo una parola.

Una riga vuota fa iniziare un nuovo
paragrafo.
```

Non ha importanza se immetti uno o tanti spazi dopo una parola.

Una riga vuota fa iniziare un nuovo paragrafo.

[3] Ci sono voci secondo cui questo è uno degli elementi chiave che saranno affrontati dal prossimo sistema LATEX3.

1.3.2 Caratteri speciali

I seguenti simboli sono caratteri riservati che hanno un significato speciale in LaTeX o non sono disponibili in tutti i *font*; quando vengono immessi nel testo, normalmente non vengono stampati, ma costringono il LaTeX ad operare in maniera indesiderata.

```
$ & % # _ { } ~ ^ \
```

Come potrete osservare, questi caratteri possono essere usati nei vostri documenti tutti allo stesso modo facendoli precedere da un *backslash*:

```
\$ \& \% \# \_ \{ \}
```

$ & % # _ { }

Gli altri simboli e molti altri ancora possono essere stampati con comandi speciali nelle formule matematiche o come accenti. Il carattere *backslash* \ *non* può essere immesso aggiungendo un altro *backslash* davanti ad esso(\\): questa sequenza si usa per le interruzioni di riga[4].

1.3.3 Comandi LaTeX

I comandi LaTeX sono *case sensitive* e accettano uno dei due seguenti formati:

- Iniziano con un backslash \ e poi hanno un nome che consiste di sole lettere; i nomi dei comandi terminano con uno spazio, un numero o un altro carattere 'non lettera'.
- Sono costituiti da un *backslash* ed esattamente da un carattere speciale.

Il LaTeX ignora lo spazio dopo i comandi. Se volete ottenere uno spazio dopo un comando, dovete inserire `{}` e un carattere di spazio o un comando speciale di spaziatura dopo il nome del comando. Le `{}` impediscono a LaTeX di divorare tutto lo spazio dopo il nome del comando.

```
Ho letto che Knuth divide la
gente che lavora con il \TeX{} in
\TeX{}nici e \TeX perti.\\
Data odierna: \today.
```

Ho letto che Knuth divide la gente che lavora con il TeX in TeXnici e TeXperti.
Data odierna: 6 marzo 2002.

Alcuni comandi necessitano di un parametro che deve essere fornito tra parentesi graffe `{ }` dopo il nome del comando. Alcuni comandi accettano parametri opzionali che si aggiungono dopo il nome del comando tra parentesi quadre `[ ]`. I prossimi esempi si servono di alcuni comandi LaTeX. Non vi preoccupate, saranno spiegati in seguito.

[4] Provate il comando `$\backslash$` invece; esso produce un '\'.

```
Puoi \textsl{appoggiarti} a me!
```

Puoi *appoggiarti* a me!

```
Per favore, inizia una nuova riga
proprio qui!\newline
Grazie!
```

Per favore, inizia una nuova riga proprio qui!
Grazie!

1.3.4 Commenti

Quando il LaTeX incontra un carattere `%` durante l'elaborazione di un file di input, ignora il resto della riga attuale, l'interruzione di riga, e tutti gli spazi bianchi all'inizio della riga successiva.

Questo può essere usato per scrivere commenti nel file di input, che non saranno mostrati nella versione stampata.

```
Ecco un % stupido
% Meglio: istruttivo <----
esempio: Supercal%
              ifragilist%
    ichespiralidoso
```

Ecco un esempio: Supercalifragilistichespira-lidoso

Il carattere `|`lunghe righe di input in cui non sono permessi spazi bianchi o interruzioni di riga.

Per commenti più lunghi si dovrebbe usare l'ambiente `comment` fornito dal pacchetto **verbatim**.

```
Ecco un altro
\begin{comment}
piuttosto stupido,
ma utile
\end{comment}
esempio per includere commenti
nel vostro documento.
```

Ecco un altro esempio per includere commenti nel vostro documento.

1.4 Struttura del file di input

Quando il LaTeX 2_ε elabora un file di input, si aspetta di seguire una certa struttura. Ogni file di input deve di conseguenza iniziare con un comando

```
\documentclass{...}
```

Questo specifica che tipo di documento si ha intenzione di scrivere. Si possono, oltre a questo, includere comandi che influenzano lo stile dell'intero documento, o si possono caricare pacchetti che aggiungono nuove caratteristiche al sistema LaTeX. Per caricare un tale pacchetto si usa il comando

```
\usepackage{...}
```

Quando tutto il lavoro di preparazione sarà compiuto[5], si inizierà con il corpo del testo con il comando

```
\begin{document}
```

Si immette ora il testo combinato ad alcuni utili comandi LaTeX. Al termine del documento si aggiunge il comando

```
\end{document}
```

che dice a LaTeX che il documento è finito; tutto ciò che segue questo comando sarà ignorato da LaTeX.

La figura 1.2 mostra i contenuti di un file LaTeX 2$_\varepsilon$ di lunghezza minima. Un file di input un po' più complicato è fornito nella figura 1.3.

```
\documentclass{article}
\begin{document}
Piccolo \'e bello.
\end{document}
```

Figura 1.2: Un file LaTeX di lunghezza minima.

1.5 Il layout del documento

1.5.1 Classi di documento

La prima informazione che il LaTeX richiede quando elabora un file di input è il tipo di documento che l'autore desidera realizzare; questo si specifica con il comando `\documentclass`.

`\documentclass[`*opzioni*`]{`*classe*`}`

Qui *classe* specifica il tipo di documento da realizzare.

La tabella 1.1 elenca le classi di documento descritte in questa introduzione. La distribuzione LaTeX 2$_\varepsilon$ fornisce classi aggiuntive per altri documenti,

[5] L'area tra `\documentclass` e `\begin{document}` è chiamata *preambolo*.

compresse lettere e lucidi. Il parametro *opzioni* adatta il comportamento della classe del documento; le opzioni devono essere separate da virgole. Le opzioni più comuni per le classi di documento standard sono elencate nella tabella 1.2.

Esempio: un file di input per un documento LaTeX potrebbe iniziare con la riga

```
\documentclass[11pt,twoside,a4paper]{article}
```

che ordina a LaTeX a impaginare il documento come *articolo* con una dimensione del *font* di base pari a *undici punti*, e a realizzare un layout adeguato alla stampa *in due facciate* in *formato A4*.

```
\documentclass[a4paper,11pt]{article}
\usepackage{latexsym}
\author{H.~Partl}
\title{Minimalismo}
\frenchspacing
\begin{document}
\maketitle
\tableofcontents
\section{Start}
Bene, qui inizia il mio grazioso articolo.
\section{End}
\ldots{} e qui finisce.
\end{document}
```

Figura 1.3: Un esempio realistico di articolo di giornale.

Tabella 1.1: Classi di documento.

`article` per articoli in riviste scientifiche, presentazioni, brevi relazioni, documentazione di programmi, inviti,...

`report` per relazioni più lunghe contenenti diversi capitoli, piccoli libri, tesine,...

`book` per libri veri e propri

`slides` per lucidi. La classe usa grosse lettere sans serif. Potreste prendere in considerazione l'uso di FoilTeX[a] al suo posto.

[a] `CTAN:/tex-archive/macros/latex/contrib/supported/foiltex`

Tabella 1.2: Opzioni delle classi di documento.

`10pt`, `11pt`, `12pt` Imposta la dimensione del *font* principale nel documento; se non è specificata, si assume `10pt`.

`a4paper`, `letterpaper`,... Definisce le dimensioni del foglio. La dimensione predefinita è `letterpaper`. Oltre a questa possono essere specificate `a5paper`, `b5paper`, `executivepaper`, e `legalpaper`.

`fleqn` Compone le formule visualizzate allineandole a sinistra invece che centrandole.

`leqno` Dispone la numerazione delle formule sulla sinistra invece che sulla destra.

`titlepage`, `notitlepage` Specfica se dopo il titolo del documento debba avere inizio o no una nuova pagina. La classe `article` non dà inizio a una nuova pagina per impostazione predefinita, mentre `report` e `book` agiscono in questo modo.

`twocolumn` Dà istruzioni a LaTeX di impaginare il documento in due colonne.

`twoside, oneside` Specifica se deve essere generato un output a doppia o singola facciata. Le classi `article` e `report` sono a singola facciata e la classe `book` è a doppia facciata per impostazione predefinita. Va notato che quest'opzione riguarda solo lo stile del documento; l'opzione `twoside` *non* comunica alla stampante usata che l'output di stampa debba effettivamente essere a due facciate.

`openright, openany` Fa iniziare i capitoli solo nella pagina destra o nella successiva pagina a disposizione. Questa opzione non funziona con la classe `article`, poiché questa non contiene informazioni riguardo ai capitoli. La classe `report` fa iniziare in maniera predefinita i capitoli sulla prossima pagina a disposizione e la classe `book` li fa iniziare sulle pagine destre.

1.5.2 I pacchetti

Non appena si scriveranno dei documenti, si scoprirà probabilmente che ci sono alcuni settori in cui il LaTeX di base non può risolvere il problema. Se si vuole includere grafica, testo a colori o codice sorgente da un file nel documento, bisognerà estendere le capacità del LaTeX; le estensioni sono dette pacchetti. I pacchetti si attivano col comando

`\usepackage[`*opzioni*`]{`*package*`}`

dove *package* è il nome del pacchetto e *opzioni* è una lista di parole chiave che danno l'avvio a speciali caratteristiche del pacchetto. Alcuni pacchetti sono inclusi nella distribuzione base di LaTeX 2_ε (consultare la tabella 1.3); altri sono forniti a parte. Si possono reperire maggiori informazioni sui pacchetti installati nel sistema nella *Local Guide* [4]. La prima fonte di informazione relativamente ai pacchetti LaTeX è *The LaTeX Companion* [3]; esso contiene le descrizioni di centinaia di pacchetti assieme a informazioni su come scrivere le proprie estensioni al LaTeX 2_ε.

1.6 I file con cui si ha a che fare

Quando si lavorerà con il LaTeX ci si troverà presto in un labirinto di file con varie estensioni e probabilmente disorientati. In basso è riportata una lista relativa ai vari tipi di file in cui ci si può imbattere lavorando con il TeX. Va notato che questa tabella non si pretende di fornire una lista esaustiva di estensioni; se qualcuno pensa che non sia presente un tipo che ritiene importante, scriva pure due righe.

.tex Input file di LaTeX o TeX. Da compilare con `latex`.

.sty Pacchetto di macro di LaTeX. Questo è un file che si può caricare nel documento LaTeX con il comando `\usepackage`.

.dtx Documented TeX. È questo il principale formato di distribuzione per i file di stile del LaTeX. Se si compila un file .dtx si ottiene del codice macro documentato relativo al pacchetto LaTeX contenuto nel file .dtx.

.ins È il file di installazione per i file contenuti nel corrispondente file .dtx. Se si scarica un pacchetto LaTeX da Internet, si otterrà normalmente un file .dtx e un file .ins; sul file .ins si esegue LaTeX per decomprimere il file .dtx.

.cls I file delle classi definiscono come apparirà il documento; sono scelti con il comando `\documentclass`.

Quando si eseguirà LaTeX sul file di input saranno creati i seguenti file:

Tabella 1.3: Alcuni pacchetti distribuiti con LaTeX.

doc Permette la documentazione di programmi LaTeX.
Descritto in `doc.dtx`[a] e in *The LaTeX Companion* [3].

exscale Fornisce versioni ridimensionate del *font* dell'estensione math.
Descritto in `ltexscale.dtx`.

fontenc Specifica quale codifica dei *font* LaTeX dovrebbe usare.
Descritto in `ltoutenc.dtx`.

ifthen Fornisce comandi nella forma
'se... allora fai... altrimenti fai...'
Descritto in `ifthen.dtx` e *The LaTeX Companion* [3].

latexsym Per accedere al *font* "symbol" di LaTeX si dovrebbe usare il pacchetto `latexsym`. Descritto in `latexsym.dtx` e in *The LaTeX Companion* [3].

makeidx Fornisce comandi per realizzare indici. Descritto nel paragrafo 4.3 e in *The LaTeX Companion* [3].

syntonly Analizza un documento senza impaginarlo.

inputenc Permette di specificare una codifica di input come ASCII, ISO Latin-1, ISO Latin-2, tabelle di codici 437/850 IBM, Apple Macintosh, Next, ANSI-Windows o una definita dall'utente. Descritto in `inputenc.dtx`.

[a] Questo file dovrebbe essere installato nel sistema, e si dovrebbe poter ottenere un file `dvi` digitando `latex doc.dtx` in una directory in cui si hanno i permessi di scrittura; lo stesso vale per tutti gli altri file citati in questa tabella.

.dvi File *device independent*. Questo è il principale risultato dell'esecuzione di LaTeX. Si può visualizzarne il contenuto con un programma di visualizzazione DVI o è possibile inviarlo alla stampante con `dvips` o un'applicazione simile.

.log Fornisce un resoconto dettagliato di cosa è avvenuto durante l'ultima esecuzione del compilatore.

.toc Conserva tutti i titoli dei paragrafi. Viene letto nella successiva esecuzione del compilatore ed è usato per realizzare l'indice.

.lof Questo è simile al .toc ma riguarda la lista delle figure.

.lot Analogamente per la lista delle tabelle.

.aux Un altro file che trasporta informazioni da un'esecuzione all'altra del compilatore; tra le altre cose, il file .aux è usato per conservare le informazioni associate ai riferimenti incrociati.

.idx È presente se il documento contiene un indice analitico. Il LaTeX memorizza tutte le parole che vanno nell'indice analitico in questo file. Questo file deve essere elaborato con `makeindex`; fare riferimento al paragrafo 4.3 a pagina 62 per altre informazioni sulla realizzazione di indici analitici.

.ind È il file .idx già elaborato, pronto ad essere incluso nel documento al prossimo ciclo di compilazione.

.ilg Un *logfile* che fornisce un resoconto su ciò che è stato compiuto da `makeindex`.

1.6.1 Stili di una pagina

Il LaTeX accetta tre combinazioni predefinite di intestazione/piè di pagina—i cosiddetti stili di pagina. Il parametro *stile* del comando

`\pagestyle{`*stile*`}`

definisce quale stile sarà utilizzato. La tabella 1.4 elenca gli stili di pagina predefiniti.

È possibile cambiare lo stile di pagina della pagina corrente con il comando

`\thispagestyle{`*stile*`}`

In *The LaTeX Companion* [3] e nel paragrafo 4.4 a pagina 64 è spiegato come si possono realizzare le proprie impostazioni di intestazione e piè di pagina.

1.7 Documenti di grandi dimensioni

Se si lavora su grossi documenti, si potrebbe voler dividere il file di input in diverse parti; il LaTeX ha due comandi che vengono in aiuto a tal fine.

`\include{`*nomefile*`}`

Questo comando si usa nel corpo del documento per inserire i contenuti di un altro *file* di nome *nomefile.tex*; va notato che LaTeX farà iniziare una nuova pagina prima di elaborare i contenuti di *nomefile.tex*.

Il secondo comando può essere usato nel preambolo; esso fa in modo che LaTeX consideri in input solo alcuni dei file di `\include`.

`\includeonly{`*nomefile*`,`*nomefile*`,`...`}`

Una volta che questo comando è eseguito nel preambolo del documento, saranno eseguiti i comandi `\include` solo per i nomi di file che sono elencati nell'argomento del comando `\includeonly`; si noti che non ci devono essere spazi tra i nomi dei file e le virgole.

Il comando `\include` inizia l'impaginazione del testo, che viene incluso in una nuova pagina. Questo è d'aiuto quando si usa `\includeonly`, poiché le interruzioni di pagina non si sposteranno, persino se alcuni dei file inclusi fossero omessi. Ciò potrebbe talvolta non risultare gradito: in tal caso si potrà usare il comando

`\input{`*filename*`}`

che includerà semplicemente il file specificato; niente di più, niente di meno.

Per far sì che LaTeX controlli velocemente il documento potrete usare il pacchetto **syntonly**; questo fa sì che LaTeX scorra il documento controllando

Tabella 1.4: Gli stili di pagina predefiniti del LaTeX.

`plain` stampa i numeri di pagina nella parte inferiore della stessa, al centro del piè di pagina. Questo è lo stile di pagina predefinito.

`headings` stampa il titolo del capitolo corrente e il numero di pagina sulla intestazione di ciascuna pagina, mentre il piè di pagina rimane vuoto (è questo lo stile usato in questo documento).

`empty` imposta la intestazione e il piè di pagina in modo che non vi sia stampato nulla.

solo la sintassi e l'uso corretto dei comandi, ma senza fornire come output un file DVI. Poiché LATEX viene eseguito più velocemente in questa modalità, si potrà risparmiare del tempo prezioso; l'uso è molto semplice:

```
\usepackage{syntonly}
\syntaxonly
```

Quando si vorrà realizzare l'impaginazione vera e propria, basterà commentare la seconda riga (facendola precedere da un simbolo di percentuale).

Capitolo 2

Composizione del testo

Dopo aver letto il capitolo precedente, dovrebbero essere noti gli elementi fondamentali che costituiscono un documento LaTeX 2_ε. Questo capitolo completerà le conoscenze necessarie per realizzare veri e propri documenti.

2.1 La struttura del testo e la lingua

Il punto essenziale nella scrittura di un testo (eccezion fatta per certa letteratura DATIC[1]), è comunicare idee, informazioni, o conoscenze al lettore, che comprenderà meglio il testo se le idee sono ben strutturate, e ne apprezzerà assai meglio la struttura se la forma tipografica rispecchia la costruzione logica e semantica del contenuto.

LaTeX si distingue da altri sistemi di composizione perché è sufficiente indicargli la struttura logica e semantica del testo, dalla quale ricava la forma tipografica secondo le regole definite nel file della classe di documento e in vari file di stile.

La più importante unità di testo in LaTeX (e in tipografia) è il paragrafo. Lo si definisce "unità di testo" perché un paragrafo è la forma tipografica che deve riflettere un singolo pensiero coerente, o un'idea. Nelle sezioni seguenti si apprenderà come si può forzare un'interruzione di riga (per esempio con \\) e la fine di un paragrafo (per esempio lasciando una riga vuota nel codice sorgente). Quindi, se si apre un nuovo discorso si deve cominciare un nuovo capoverso, o altrimenti bisogna solamente andare a capo. Quando si è incerti sulla divisione dei paragrafi, si pensi al testo come veicolo di idee e pensieri. Se si chiude un paragrafo ma continua il pensiero precedente, la divisione deve essere eliminata. Se nello stesso paragrafo si introduce una linea di pensiero completamente nuova, allora occorre introdurla con un nuovo capoverso.

[1] Diversi A Tutti I Costi, traduzione dallo svizzero tedesco UVA (*Um's Verrecken Anders*).

La maggior parte delle persone sottovaluta completamente l'importanza di paragrafi ben strutturati. Molti ignorano persino qual è il significato della suddivisione in paragrafi o, specialmente in LaTeX, cominciano un nuovo paragrafo senza rendersene conto. Quest'ultimo errore è particolarmente facile da commettere se nel testo si usano equazioni. Osservando gli esempi che seguono, si cerchi di capire perché a volte si usano righe vuote (fine capoverso) prima e dopo l'equazione, e altre volte no. (Se non si comprendono ancora tutti i comandi abbastanza bene per capire questi esempi, si legga interamente questo capitolo e il seguente, e poi si ritorni nuovamente a questa sezione.)

```
% Esempio 1
\ldots quando Einstein propose la definizione
\begin{equation}
  E = m \cdot \mathrm{c}^2 \; ,
\end{equation}
che \`e allo stesso tempo la pi\`u nota
e la meno ben compresa formula della Fisica.

% Esempio 2
\ldots da cui segue la Legge di Kirchhoff sulle
correnti:
\begin{equation}
  \sum_{k=1}^{n} I_k = 0 \; .
\end{equation}

La Legge di Kirchhoff sulle tensioni pu\`o essere
ricavata \ldots

% Esempio 3
\ldots che ha alcuni vantaggi.

\begin{equation}
  I_\mathrm{D} = I_\mathrm{F} - I_\mathrm{R}
\end{equation}
costituisce la parte centrale di un modello molto
diverso di transistor. \ldots
```

L'unità di testo immediatamente più piccola è il periodo. Nei testi in lingua inglese[2], dopo un punto che conclude un periodo c'è più spazio di quanto ce ne sia dopo uno che termina un'abbreviazione. LaTeX cerca di capire quale sia il punto che si intendeva utilizzare. Se LaTeX sbaglia bisogna fornirgli l'indicazione corretta, come è spiegato più avanti in questo capitolo.

[2] E sovente anche nei testi in lingua italiana. [N.d.T.]

La strutturazione del testo si estende anche alle parti di un periodo. Le lingue, nella maggior parte, hanno regole di punteggiatura piuttosto complicate, ma per molte (comprese quelle tedesca e inglese[3]) è sufficiente ricordare cosa rappresenta la virgola: una breve pausa nel discorso. Se non si è sicuri del posizionamento di una virgola, si legga la frase ad alta voce, prendendo un breve respiro ad ogni virgola. Se questo in qualche punto dà la sensazione di essere inopportuno, la virgola va eliminata. Se in qualche altro punto si sente la necessità di prendere fiato (o di fare una breve pausa), va inserita una virgola.

Infine, i paragrafi in un testo devono essere strutturati logicamente ad un livello più alto, inserendoli in capitoli, sezioni, sottosezioni, e così via. In ogni caso l'effetto tipografico di scrivere per esempio `\section{La struttura del testo e la lingua}` è a tal punto evidente che l'uso di queste strutture ad alto livello si spiega quasi da sé.

2.2 Chiusura di riga e di pagina

2.2.1 Paragrafi giustificati

I libri sono spesso composti con tutte le righe della stessa lunghezza. LaTeX inserisce le necessarie interruzioni di riga e gli spazi tra le parole ottimizzando il contenuto dell'intero paragrafo. Inoltre, se necessario, divide le parole che non stanno agevolmente su una riga. Il modo in cui sono composti i paragrafi dipende dalla classe di documento. Normalmente la prima riga di un paragrafo è evidenziata con una rientranza, e non c'è spazio aggiuntivo tra i paragrafi. Si consulti la sezione 5.3.2 per ulteriori ragguagli.

In casi particolari può essere necessario ordinare a LaTeX di interrompere una riga. I comandi

`\\` oppure `\newline`

cominciano una nuova riga senza iniziare un nuovo paragrafo.

Il comando

`\\*`

impedisce inoltre l'interruzione di pagina dopo l'interruzione di riga.

[3] E anche la lingua italiana. [N.d.T.]

Il comando

`\newpage`

comincia una nuova pagina.

`\linebreak[`*n*`]`, `\nolinebreak[`*n*`]`, `\pagebreak[`*n*`]` e `\nopagebreak[`*n*`]`

fanno quel che dicono i loro nomi[4]. Questi comandi consentono all'autore di modificare la loro azione con l'argomento opzionale *n*, che può essere impostato ad un numero compreso tra zero e quattro. Impostando *n* a un valore inferiore a 4 si lascia a LaTeX la possibilità di ignorare il comando se il risultato dovesse essere pessimo. I comandi per interrompere ("*break*") una riga o una pagina non vanno confusi con quelli che cominciano una nuova riga o una nuova pagina ("*new*"). Quando si usa un comando di interruzione, LaTeX cerca anche di pareggiare il margine destro della pagina e la lunghezza totale della pagina, come descritto nella prossima sezione. Se davvero si vuole cominciare una nuova riga ("*new line*"), allora bisogna usare il comando corrispondente. Si lascia al lettore dire quale sia.

LaTeX cerca di produrre sempre le migliori interruzioni di riga possibili. Se non riesce a trovare il modo di spezzare le righe secondo i suoi severi criteri, lascia che la riga fuoriesca dal margine destro. LaTeX allora si lamenta ("Overfull hbox") mentre elabora il file in ingresso. Questo accade per lo più quando LaTeX non riesce a trovare un punto adatto per spezzare una parola[5]. È possibile dare istruzione a LaTeX di adottare un criterio meno rigido con il comando `\sloppy`, che evita le righe eccessivamente lunghe aumentando la spaziatura tra le parole—anche se il risultato finale non è ottimale. In questo caso l'utente viene avvisato con un messaggio ("Underfull hbox"). Nella maggior parte di tali casi il risultato non ha un bell'aspetto. Il comando `\fussy` ripristina le impostazioni predefinite di LaTeX.

2.2.2 Divisione delle parole in fin di riga

LaTeX divide le parole ogniqualvolta sia necessario. Se l'algoritmo di sillabazione non trova i punti di divisione corretti, si può porre rimedio al problema usando i seguenti comandi per dire a TeX di fare un'eccezione.

[4] `\linebreak` e `\nolinebreak` rispettivamente forzano o impediscono un'interruzione di riga, e analogamente `\pagebreak` e `\nopagebreak` per le interruzione di pagina. [N.d.T.]

[5] Benché LaTeX avverta l'utente quando ciò accade ("Overfull hbox"), le righe a cui si riferisce non sempre sono facili da trovare. Usando l'opzione `draft` nel comando `\documentclass`, queste righe saranno evidenziate con una spessa linea nera sul margine destro.

Il comando

`\hyphenation{`*elenco di parole*`}`

fa sì che le parole elencate nell'argomento siano divise solo in corrispondenza dei punti marcati da "`-`". L'argomento del comando deve contenere parole costituite solo da normali lettere o per meglio dire segni che sono considerati come normali lettere nel contesto attivo. Le indicazioni per la sillabazione sono applicate alla lingua che è attiva nel momento in cui il comando viene usato. Questo significa che il comando posto nel preambolo del documento influenzerà la sillabazione della lingua inglese. Se lo si chiama dopo `\begin{document}` e si sta usando qualche pacchetto per la localizzazione come **babel**, allora saranno attive le regole di sillabazione per la lingua selezionata tramite **babel**.

L'esempio che segue permette che "sillabazione" sia divisa, come pure "Sillabazione", e impedisce del tutto che "FORTRAN", "Fortran" e "fortran" siano divise. Nell'argomento non sono ammessi caratteri speciali o simboli.

Esempio:

```
\hyphenation{FORTRAN Sil-la-ba-zio-ne}
```

Il comando `\-` inserisce in una parola un trattino discrezionale, che diventa anche l'unico punto di divisione consentito in quella parola. Si tratta di un comando particolarmente utile per parole contenenti caratteri speciali (per esempio lettere accentate), perché LaTeX non le divide automaticamente.

```
Penso che questo sia: su\-per\-%
ca\-li\-fra\-gi\-lis\-ti\-che\-%
spi\-ra\-li\-do\-so
```

Penso che questo sia: supercalifragilistiche-
spiralidoso

Più parole possono essere mantenute insieme sulla stessa riga con il comando

`\mbox{`*testo*`}`

che fa in modo che il suo argomento sia mantenuto insieme in qualunque caso.

```
Il mio numero di telefono presto
cambier\'a. Sar\'a \mbox{0116 291
2319}.

Il parametro
\mbox{\emph{nomefile}} deve
contenere il nome del file.
```

Il mio numero di telefono presto cambierà. Sarà 0116 291 2319.

Il parametro *nomefile* deve contenere il nome del file.

2.3 Stringhe già pronte

In alcuni degli esempi nelle pagine precedenti si sono visti vari comandi LaTeX molto semplici per comporre particolari stringhe di testo:

Comando	Esempio	Descrizione
`\today`	6 marzo 2002	La data odierna nella lingua corrente
`\TeX`	TeX	Il sistema preferito di composizione testi
`\LaTeX`	LaTeX	Il nome del gioco
`\LaTeXe`	LaTeX 2_ε	L'attuale versione di LaTeX

2.4 Caratteri speciali e simboli

2.4.1 Virgolette

Per le virgolette *non* bisogna usare il carattere `"` come invece si farebbe su una macchina da scrivere. Nell'editoria ci sono speciali virgolette di apertura e di chiusura. In LaTeX, bisogna usare due ` come virgolette di apertura e due ' come virgolette di chiusura.

```
``Premere il tasto `x'.''
```

"Premere il tasto 'x'."

2.4.2 Tratti

LaTeX distingue quattro tipi di tratto. Tre di questi si ottengono con un diverso numero di trattini consecutivi. Il quarto in realtà non è neppure un tratto ma il segno matematico meno:

```
Stratford-on-Avon, e-mail\\
pagine 13--67\\
s\`\i---oppure no? \\
$0$, $1$ e $-1$
```

Stratford-on-Avon, e-mail
pagine 13–67
sì—oppure no?
0, 1 e -1

I loro nomi sono: '-' trattino (*hyphen*), '–' trattino medio (*en-dash*), '—' tratto (*em-dash*) e '$-$' meno.

2.4.3 Tilde ($\sim$)

La tilde è un carattere che si vede spesso negli indirizzi web. Per generarlo in LaTeX si può usare `\~` ma il risultato ˜ non è esattamente quel che si desidera. Si può invece provare questo:

```
http://www.rich.edu/\~{}bush \\
http://www.clever.edu/$\sim$demo
```

http://www.rich.edu/˜bush
http://www.clever.edu/∼demo

2.4.4 Punti di sospensione (...)

Su una macchina da scrivere una virgola o un punto occupano lo stesso spazio di ogni altra lettera. Nella stampa questi caratteri occupano solo un piccolo spazio e sono posti molto vicino alla lettera che li precede. Quindi non si possono inserire dei punti di sospensione battendo semplicemente tre punti, perché la spaziatura sarebbe errata. Per di più esiste un comando apposito:

```
\ldots
```

```
Non cos\`\i{} ... ma cos\`\i:\\
New York, Tokyo, Budapest,\ldots
```

Non così ... ma così:
New York, Tokyo, Budapest,...

2.4.5 Legature

Alcune combinazioni di lettere non sono composte affiancando semplicemente le differenti lettere una dopo l'altra ma usando per l'appunto simboli speciali.

ﬀ ﬁ ﬂ ﬃ... invece di ff fi fl ffi ...

È possibile evitare queste cosiddette legature inserendo un comando `\mbox{}` tra le lettere in questione. Questo potrebbe essere necessario con parole composte.

```
Non offline\\
ma off\mbox{}line
```

Non oﬄine
ma off‌line

2.4.6 Accenti e caratteri speciali

LaTeX consente l'uso degli accenti e dei caratteri speciali usati in molte lingue. La tabella 2.1 mostra ogni sorta di accento applicato alla lettera o. Naturalmente funziona anche con altre lettere.

Per mettere un accento sopra una 'i' oppure una 'j', bisogna rimuovere i puntini. Si può far questo scrivendo `\i` e `\j`.

```
H\^otel, na\"\i ve, \'el\`eve,\\
sm\o rrebr\o d, !`Se\~norita!,\\
Sch\"onbrunner Schlo\ss{}
Stra\ss e
```

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

Tabella 2.1: Accenti e caratteri speciali.

ò	`\`o`	ó	`\'o`	ô	`\^o`	õ	`\~o`
ō	`\=o`	ȯ	`\.o`	ö	`\"o`	ç	`\c c`
ŏ	`\u o`	ǒ	`\v o`	ő	`\H o`	ǫ	`\c o`
ọ	`\d o`	o̲	`\b o`	o͡o	`\t oo`		
œ	`\oe`	Œ	`\OE`	æ	`\ae`	Æ	`\AE`
å	`\aa`	Å	`\AA`				
ø	`\o`	Ø	`\O`	ł	`\l`	Ł	`\L`
ı	`\i`	ȷ	`\j`	¡	`` !` ``	¿	`` ?` ``

2.5 LaTeX multilingue

Se si vuole scrivere documenti in lingue diverse dall'inglese, ci sono due ambiti per cui LaTeX deve essere opportunamente configurato:

1. Tutte le stringhe di testo generate automaticamente[6] devono essere adattate alla nuova lingua. Per molte lingue, si possono compiere queste modifiche usando il pacchetto babel di Johannes Braams.

2. LaTeX deve conoscere le regole di sillabazione per la nuova lingua. Far usare le regole di sillabazione a LaTeX è più difficoltoso. Bisogna ricostruire il file di formato (*format file*) con differenti schemi di sillabazione abilitati. La *Local Guide* [4] dovrebbe fornire maggiori informazioni a questo proposito.

Se il proprio sistema è già appropriatamente configurato, si può attivare il pacchetto babel aggiungendo il comando

```
\usepackage[lingua]{babel}
```

dopo il comando `\documentclass`. Le lingue che il proprio sistema consente di usare dovrebbero anche essere elencate nella *Local Guide* [4]. Babel

[6] Indice, Elenco delle figure, ...

attiverà automaticamente le regole di sillabazione appropriate per la lingua che è stata scelta. Se il proprio formato LaTeX non consente la sillabazione nella lingua scelta, babel funzionerà ancora ma disabiliterà la divisione delle parole in fin di riga, il che ha un effetto piuttosto negativo sull'aspetto del documento.

Per alcune lingue, babel definisce anche nuovi comandi che semplificano l'immissione di caratteri speciali. La lingua tedesca, per esempio, contiene molte dieresi (äöü). Con babel, si può immettere una ö battendo `"o` invece di `\"o`.

Alcuni sistemi consentono di immettere caratteri speciali direttamente dalla tastiera. LaTeX è in grado di gestirli. A partire dalla versione di LaTeX 2_ε del dicembre 1994, nella distribuzione di base consente l'uso di diverse codifiche, grazie al pacchetto inputenc. Quando si usa questo pacchetto, va considerato che altre persone potrebbero non essere in grado di visualizzare correttamente i file così realizzati, perché utilizzano di una diversa codifica. Su un PC, per esempio, il simbolo tedesco ä è codificato come 132, ma su alcuni sistemi Unix che usano ISO-LATIN 1 è codificato come 228. Bisogna dunque usare questa possibilità con attenzione.

La codifica dei *font* è una questione diversa. Essa definisce in quale posizione è collocata ogni lettera all'interno di un *TeX-font*. I *font* TeX Computer Modern originali contengono solo i 128 caratteri del vecchio set ASCII 7-bit. Quando sono richiesti caratteri accentati, TeX li crea combinando un normale carattere con un accento. Benché il risultato sia perfetto, questo approccio impedisce il funzionamento della sillabazione automatica all'interno di parole contenenti caratteri accentati.

Fortunatamente, la maggior parte delle distribuzioni recenti di TeX include una copia dei *font* EC, che hanno lo stesso aspetto dei Computer Modern ma contengono caratteri speciali per la maggior parte delle lettere accentati usate nelle lingue europee. Usando questi *font* si può migliorare la sillabazione in documenti non in lingua inglese. I *font* EC sono attivati includendo il pacchetto fontenc nel preambolo del documento.

```
\usepackage[T1]{fontenc}
```

2.6 Lo spazio tra le parole

Per ottenere il margine destro giustificato, LaTeX inserisce tra le parole spazi di dimensione variabile. Alla fine di una frase inserisce uno spazio leggermente maggiore, perché questo rende il testo più leggibile. LaTeX presuppone che i periodi si chiudano con un punto, un punto interrogativo, o un punto esclamativo. Se un punto segue una lettera maiuscola non viene interpretato come conclusione di una frase, perché normalmente un punto che viene dopo una lettera maiuscola si trova nelle abbreviazioni.

Ogni eccezione a queste regole generali deve essere specificata dall'autore. Il comando `\` prima di uno spazio produce uno spazio che non può essere allargato. Un carattere tilde '`~`' produce uno spazio che non può essere allargato e inoltre impedisce un'interruzione di riga. Il comando `\@` davanti a un punto specifica che quel punto conclude una frase anche se si trova dopo una lettera maiuscola.

```
Il Sig.~Rossi fu lieto
di vederla.\\
cf.~Fig.~5\\
Mi piace il BASIC\@. E a te?
```

Il Sig. Rossi fu lieto di vederla.
cf. Fig. 5
Mi piace il BASIC. E a te?

Lo spazio aggiuntivo dopo un punto può essere disabilitato con il comando

```
\frenchspacing
```

che dice a LaTeX di *non* inserire dopo un punto più spazio di quanto sia inserito dopo ogni normale carattere. Questo è molto comune in lingue diverse dall'inglese, fuorché nelle bibliografie. Se si usa `\frenchspacing`, il comando `\@` non è più necessario.

2.7 Titoli, capitoli, e sezioni

Per aiutare il lettore ad orientarsi nel testo, bisogna dividere il documento in capitoli, sezioni, e sottosezioni. LaTeX permette di farlo con speciali comandi che prendono come argomento il relativo titolo. Spetta all'autore usarli nell'ordine corretto.

Per la classe `article` sono disponibili i seguenti comandi di sezionamento del testo:

```
\section{...}           \paragraph{...}
\subsection{...}        \subparagraph{...}
\subsubsection{...}     \appendix
```

Con le classi `report` e `book` si possono usare due comandi di sezionamento in più:

```
\part{...}              \chapter{...}
```

Dal momento che la classe `article` non sa nulla dei capitoli, è piuttosto semplice raggruppare degli articoli come capitoli di un libro. La spaziatura tra sezioni, la numerazione e il corpo usato nei titoli sono impostati automaticamente da LaTeX.

Due dei comandi di sezionamento sono un poco speciali:

- Il comando `\part` non influisce sulla numerazione dei capitoli.
- Il comando `\appendix` non richiede alcun argomento. Cambia semplicemente la numerazione dei capitoli in lettere[7].

LaTeX crea l'indice prendendo i titoli delle sezioni e i numeri di pagina dall'ultimo ciclo di compilazione del documento. Il comando

`\tableofcontents`

produce l'indice nel punto in cui è inserito. Un documento deve essere compilato (cioè elaborato da LaTeX) due volte per ottenere un indice corretto. Talvolta potrebbe essere necessario compilare il documento una terza volta. LaTeX avvisa quando questo è necessario.

Di tutti i comandi di sezionamento elencati esiste anche una versione "asterisco". La versione "asterisco" di un comando è costituita da un asterisco `*` aggiunto alla fine del comando, e genera titoli che non compaiono nell'indice e non sono numerati. Il comando `\section{Aiuto}`, per esempio, diventerebbe `\section*{Aiuto}`.

Normalmente i titoli appaiono nell'indice esattamente come sono nel testo. A volte questo non è possibile, perché sono troppo lunghi per trovarvi posto. Si può specificare la voce per l'indice come argomento opzionale prima del vero titolo.

```
\chapter[Leggilo! \`E emozionante]{Questo \`e un
   titolo lunghissimo e particolarmente noioso}
```

Il comando

`\maketitle`

crea il titolo del documento. Gli elementi che costituiscono il titolo devono essere definiti dai comandi

`\title{...}`, `\author{...}` e (opzionale) `\date{...}`

prima di chiamare `\maketitle`. Come argomento di `\author` sono ammessi più nomi separati dal comando `\and`.

Nella figura 1.3 a pagina 8 si può trovare un esempio di alcuni dei comandi sopra citati.

[7] Per la classe `article` cambia la numerazione delle sezioni.

Oltre ai comandi di sezionamento illustrati sopra, LaTeX 2_ε ha introdotto tre nuovi comandi da usare con la classe `book`.

`\frontmatter`, `\mainmatter` e `\backmatter`

Questi ultimi sono utili per suddividere le proprie pubblicazioni; modificano le intestazioni e la numerazione delle pagine in modo che appaiano come ci si aspetterebbe in un libro.

2.8 Riferimenti incrociati

Nei libri, *report* e articoli, si trovano spesso dei riferimenti incrociati a figure, tabelle e particolari porzioni di testo. LaTeX fornisce i seguenti comandi per i riferimenti incrociati

`\label{`*segnalibro*`}`, `\ref{`*segnalibro*`}` e `\pageref{`*segnalibro*`}`

dove *segnalibro* è un identificatore scelto dall'utente. LaTeX sostituisce a `\ref` il numero della sezione, sottosezione, figura, tabella, o teorema dopo il quale si trova il corrispondente comando `\label`. Il comando `\pageref` stampa il numero di pagina del corrispondente comando `\label`[8]. Proprio come per i titoli di sezione, vengono usati i numeri relativi all'ultima compilazione.

```
Ecco un riferimento a questa
sezione\label{sec:this}:
``vedi sezione~\ref{sec:this} a
pagina~\pageref{sec:this}''.
```

Ecco un riferimento a questa sezione: "vedi sezione 2.8 a pagina 28".

2.9 Note a piè di pagina

Il comando

`\footnote{`*testo della nota a piè di pagina*`}`

stampa una nota a piè di pagina in fondo alla pagina corrente. Le note dovrebbero sempre essere poste dopo la parola[9] o la frase a cui si riferiscono[10].

---

[8] Si noti che questi comandi non sanno a che cosa fanno riferimento. Il comando `\label` si limita a salvare l'ultimo numero generato automaticamente.

[9] In questo caso dopo una parola.

[10] In italiano le note a piè di pagina relative a una frase o a una sua parte si trovano prima del segno di interpunzione.

```
Le note a pi\`e di
pagina\footnote{Questo ne \`e
  un esempio.} sono spesso
usate dagli utenti di \LaTeX.
```

Le note a piè di pagina[a] sono spesso usate dagli utenti di LaTeX.

[a]Questo ne è un esempio.

2.10 Parole enfatizzate

Quando si batte a macchina, `le parole importanti sono enfatizzate` `sottolineandole`. Nei libri, invece, le parole sono enfatizzate stampandole in *corsivo*. LaTeX fornisce il comando

`\emph{`*testo*`}`

per enfatizzare il testo. Ciò che effettivamente il comando fa al suo argomento dipende dal contesto:

```
\emph{Se si enfatizza
  all'interno di un testo
  enfatizzato, \LaTeX{} usa
  il \emph{font normale}
  per enfatizzare.}
```

Se si enfatizza all'interno di un testo enfatizzato, LaTeX usa il font normale *per enfatizzare.*

Si osservi la differenza tra dire a LaTeX di *enfatizzare* qualcosa e dirgli di usare un *font* diverso:

```
\textit{Si pu\`o
  \emph{enfatizzare} del
  testo se \`e in corsivo,}
\textsf{in un \emph{font}
  senza grazie,}
\texttt{o nei \emph{font}
  a larghezza fissa.}
```

Si può enfatizzare *del testo se è in corsivo,* in un *font* senza grazie, `o nei` *`font`* `a larghezza fissa.`

2.11 Ambienti

Un generico ambiente si invoca con

`\begin{`*ambiente*`}` *testo* `\end{`*ambiente*`}`

dove *ambiente* è il nome dell'ambiente (*environment*). Gli ambienti possono essere chiamati diverse volte l'uno all'interno dell'altro, purché l'ordine di chiamata venga rispettato.

```
\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}
```

Nelle seguenti sezioni vengono illustrati i principali ambienti.

2.11.1 Elenchi puntati, numerati e descrizioni

L'ambiente `itemize` è adatto per piccole liste (elenchi puntati), l'ambiente `enumerate` per elenchi numerati, e l'ambiente `description` per descrizioni.

```
\flushleft
\begin{enumerate}
\item Gli ambienti
introdotti possono essere
messi insieme a piacere:
\begin{itemize}
\item Ma potrebbero
cominciare a sembrare
stupidi.
\item[-] Con una lineetta.
\end{itemize}
\item Quindi attenzione:
\begin{description}
\item[Cose] stupide non
diventeranno intelligenti
solo perch\'e sono
in una lista.
\item[Concetti]
intelligenti, tuttavia,
possono essere presentati
elegantemente in una lista.
\end{description}
\end{enumerate}
```

1. Gli ambienti introdotti possono essere messi insieme a piacere:
   - Ma potrebbero cominciare a sembrare stupidi.
   - \- Con una lineetta.
2. Quindi attenzione:

   Cose stupide non diventeranno intelligenti solo perché sono in una lista.

   Concetti intelligenti, tuttavia, possono essere presentati elegantemente in una lista.

2.11.2 Centrare e allineare i paragrafi

Gli ambienti `flushleft` e `flushright` generano paragrafi che sono allineati a sinistra o a destra. L'ambiente `center` genera testo centrato. Se le linee non vengono esplicitamente interrotte dal comando \\, LaTeX andrà a capo automaticamente.

```
\begin{flushleft}
Questo testo \`e\\ allineato a
sinistra. \LaTeX{} non cerca
di creare linee di uguale
lunghezza.
\end{flushleft}
```

Questo testo è
allineato a sinistra. LaTeX non cerca di creare linee di uguale lunghezza.

```
\begin{flushright}
Questo testo \'e allineato a\\
destra. \LaTeX{} non cerca
di creare linee di uguale
lunghezza.
\end{flushright}
```

Questo testo è allineato a
destra. LaTeX non cerca di creare linee di
uguale lunghezza.

```
\begin{center}
Al centro\\della Terra
\end{center}
```

Al centro
della Terra

2.11.3 Citazioni e versi

L'ambiente `quote` è utile per citazioni brevi, frasi importanti ed esempi.

```
Una regola pratica per la
lunghezza delle linee \`e:
\begin{quote}
Nessuna linea dovrebbe contenere
pi\`u di 66~caratteri.
\end{quote}
Ecco perch\'e le pagine in \LaTeX{}
hanno di default margini cos\`\i{}
larghi, e nei giornali si usano
spesso pi\`u colonne.
```

Una regola pratica per la lunghezza delle linee è:

> Nessuna linea dovrebbe contenere più di 66 caratteri.

Ecco perché le pagine in LaTeX hanno di default margini così larghi, e nei giornali si usano spesso più colonne.

Ci sono due ambienti simili: `quotation` e `verse`. L'ambiente `quotation` è utile per citazioni lunghe, formate da più paragrafi, perché non li indenta. L'ambiente `verse` si usa per le poesie, dove la divisione delle linee è importante. I versi sono separati attraverso il comando `\\` alla fine di ogni linea.

```
Conosco una sola poesia a memoria.
Riguarda un pachiderma.
\begin{flushleft}
\begin{verse}
Un elefante si dondolava\\
sopra il filo di una
ragnatela.\\
Trovando il gioco molto
interessante\\
and\`o a chiamare un
altro elefante.
\end{verse}
\end{flushleft}
```

Conosco una sola poesia a memoria. Riguarda un pachiderma.

Un elefante si dondolava
sopra il filo di una ragnatela.
Trovando il gioco molto
interessante
andò a chiamare un altro
elefante.

2.11.4 Stampare alla lettera

Il testo racchiuso tra `\begin{verbatim}` e `\end{verbatim}` verrà stampato direttamente, come se fosse battuto a macchina, con tutti gli spazi e le interruzioni di linea, senza che LaTeX interpreti ed esegua comandi.

All'interno di un paragrafo, un simile effetto può essere ottenuto con

`\verb+`*testo*`+`

Il `+` è solo un esempio di carattere delimitatore. Si può usare qualsiasi carattere tranne le lettere, `*` o lo spazio. Molti esempi di LaTeX in questo opuscolo sono realizzati con questo comando.

```
Il comando \verb|\ldots|\ldots

\begin{verbatim}
10 PRINT "CIAO A TUTTI ";
20 GOTO 10
\end{verbatim}
```

Il comando `\ldots`...

```
10 PRINT "CIAO A TUTTI ";
20 GOTO 10
```

```
\begin{verbatim*}
la versione asteriscata
dell'ambiente
verbatim     enfatizza
gli spazi   nel testo
\end{verbatim*}
```

```
la␣versione␣asteriscata
dell'ambiente
verbatim␣␣␣␣␣enfatizza
gli␣spazi␣␣␣nel␣testo
```

Il comando `\verb` può essere usato similmente con un asterisco:

```
\verb*|cos\'\i{}  :-) |
```

```
cos\'\i{}␣␣:-)␣
```

L'ambiente `verbatim` e il comando `\verb` non possono essere usati in parametri di altri comandi.

2.11.5 Tabelle

L'ambiente `tabular` può essere usato per comporre splendide tabelle opzionalmente corredate da linee orizzontali e verticali. LaTeX determina automaticamente l'ampiezza delle colonne.

L'argomento *formato* del comando

`\begin{tabular}{`*formato*`}`

definisce il formato della tabella. Si usa una `l` per specificare una colonna con testo allineato a sinistra (*left*), `r` per testo allineato a destra (*right*), e `c` per testo centrato (*centered*); `p{`*larghezza*`}` per una colonna contenente testo giustificato in un paragrafo, e `|` per una linea verticale.

All'interno di un ambiente `tabular`, `&` salta alla colonna successiva, `\\` inizia una nuova riga e `\hline` inserisce una linea orizzontale.

```
\begin{tabular}{|r|l|}
\hline
7C0 & esadecimale \\
3700 & ottale \\
11111000000 & binario \\
\hline \hline
1984 & decimale \\
\hline
\end{tabular}
```

7C0	esadecimale
3700	ottale
11111000000	binario
1984	decimale

```
\begin{tabular}{|p{4.7cm}|}
\hline
Benvenuti al paragrafo riquadrato.
Speriamo vivamente che sia
di vostro gradimento.\\
\hline
\end{tabular}
```

Benvenuti al paragrafo riquadrato. Speriamo vivamente che sia di vostro gradimento.

Il separatore di colonna può essere specificato con il costrutto `@{...}`. Questo comando elimina lo spazio tra le colonne e lo sostituisce con qualsiasi cosa stia tra le parentesi graffe. Un utilizzo comune di questo comando è illustrato di seguito col problema dell'allineamento dei decimali. Un'altra possibile applicazione è la soppressione dello spazio tra le colonne di una tabella con `@{}`.

```
\begin{tabular}{@{} l @{}}
\hline
senza spazi\\
\hline
\end{tabular}
```

senza spazi

```
\begin{tabular}{l}
\hline
spazi a dritta e a manca\\
\hline
\end{tabular}
```

spazi a dritta e a manca

Dal momento che non ci sono comandi predefiniti per allineare colonne numeriche alla virgola decimale[11], possiamo "barare" e ottenere il risultato

[11] Il pacchetto `dcolumn` potrebbe comunque essere utile.

usando due colonne: una allineata a destra per la parte intera e una allineata a sinistra per i decimali. Il comando `@{,}` nel *formato* dell'ambiente `tabular` sostituisce la normale spaziatura tra le colonne con una semplice ",", dando l'impressione di un'unica colonna giustificata dalla virgola decimale. Nei numeri, naturalmente, la virgola decimale va sostituita con un separatore di colonna (`&`). È possibile specificare un'etichetta in una colonna posta sopra questa "colonna" di numeri usando il comando `\multicolumn`.

```
\begin{tabular}{c r @{,} l}
Espressione       &
\multicolumn{2}{c}{Valore} \\
\hline
$\pi$               & 3&1416  \\
$\pi^{\pi}$         & 36&46   \\
$(\pi^{\pi})^{\pi}$ & 80662&7 \\
\end{tabular}
```

Espressione	Valore
π	3,1416
π^π	36,46
$(\pi^\pi)^\pi$	80662,7

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}
{\textbf{Ambarab\`a}} \\
\hline
Cicc\`\i{} & Cocc\`o! \\
\hline
\end{tabular}
```

Ambarabà	
Ciccì	Coccò!

2.12 Oggetti mobili

Molte pubblicazioni oggi contengono figure e tabelle. Questi elementi richiedono un trattamento particolare, perché non possono essere spezzate e stampate su più pagine. Un metodo potrebbe essere quello di cominciare una nuova pagina ogni volta che una figura o una tabella è troppo grande per stare sulla pagina corrente. Questo approccio lascerebbe alcune pagine parzialmente vuote, e il risultato sarebbe insoddisfacente.

La soluzione a questo problema è di rendere mobili (o flottanti, *floating*) le figure o le tabelle che non stanno sulla pagina corrente e stamparle su una pagina successiva, riempiendo quella attuale con il testo successivo. LaTeX offre due ambienti per gli oggetti mobili, uno per le tabelle e uno per le figure. Per ottenere il massimo da questi due ambienti è importante comprendere almeno approssimativamente come LaTeX tratta internamente gli oggetti mobili. In caso contrario, essi possono diventare una fonte di frustrazione, perché LaTeX non li mette mai dove ci si aspetterebbe.

Verranno dapprima esaminati i comandi che LaTeX fornisce per gli oggetti mobili.

Tutto il materiale presente in un ambiente `figure` o `table` verrà trattato come oggetto mobile. Entrambi gli ambienti prevedono un parametro opzionale

`\begin{figure}[`*posizionamento*`]` o `\begin{table}[`*posizionamento*`]`

chiamato *posizionamento*. Questo parametro serve a dire a LaTeX dove gli oggetti possono essere spostati. Un *posizionamento* è costituito da una stringa di *caratteri di trasferimento* (tabella 2.2).

Tabella 2.2: Caratteri di trasferimento.

Carattere	Permesso di spostare l'oggetto...
`h`	qui (*here*), nel punto esatto in cui compare nel testo. Si usa principalmente per piccoli oggetti mobili.
`t`	in cima (*top*) a una pagina
`b`	in fondo (*bottom*) a una pagina
`p`	in una pagina speciale (*page*) contenente solo oggetti mobili.
`!`	senza considerare molti dei parametri interni[a] che potrebbero impedire a questo oggetto di essere spostato.

[a]Come il numero massimo di oggetti mobili che è possibile avere in una pagina.

Nota: `0pt` e `1.05em` sono unità del TeX. Si può leggere di più su questo argomento nella tabella refunits a pagina 76.

Una tabella potrebbe ad esempio iniziare con la seguente linea

```
\begin{table}[!hbp]
```

Il posizionamento `[!hbp]` consente a LaTeX di mettere la tabella esattamente qui (`h`) o in fondo a qualche pagina (`b`) o in una pagina di soli oggetti mobili (`p`), e tutto questo anche se il risultato non è eccezionale (`!`). Se il posizionamento non viene specificato, le classi standard assumono che sia `[tbp]`.

LaTeX posizionerà ogni oggetto mobile che incontra, secondo il posizionamento specificato dall'autore. Se un oggetto non può essere posizionato nella pagina corrente, viene accumulato nella coda delle *figure* o delle *tabelle*[12]. Quando viene iniziata una nuova pagina, LaTeX controlla innanzitutto se è possibile una pagina speciale di soli oggetti mobili con quelli presenti nelle code. Se questo non è possibile, il primo oggetto in ciascuna coda è trattato come se fosse appena comparso nel testo: LaTeX prova ancora a posizionarlo secondo il suo posizionamento (tranne per '`h`', che non è più possibile). Tutti i nuovi oggetti mobili che compaiono nel testo vengono posti nelle rispettive code. LaTeX mantiene strettamente l'ordine di apparizione

[12]Queste code sono di tipo FIFO (*First In Fisrt Out*, primo arrivato primo servito).

originale per ogni tipo di oggetto mobile. Ecco perché una figura che non può essere posizionata spinge tutte le figure successive alla fine del documento. Quindi:

> Se LaTeX non mette gli oggetti mobili dove ci si aspettava, è spesso a causa di un solo oggetto che blocca una delle due code.

Dopo aver spiegato la parte difficile, rimangono alcune altre cose da menzionare riguardo gli ambienti `table` e `figure`. Con il comando

`\caption{`*caption text*`}`

è possibile specificare una didascalia per l'oggetto mobile. LaTeX aggiunge la stringa "Figura" o "Tabella" e un numero progressivo.

I due comandi

`\listoffigures` e `\listoftables`

agiscono analogamente al comando `\tableofcontents`, stampando una lista di figure o tabelle, rispettivamente. In queste liste, verrà ripetuta tutta la didascalia. Se si ha la tendenza a usare didascalie lunghe, è consigliabile fornirne una versione più corta che comparirà nelle liste. Questo si ottiene specificando la versione breve tra parentesi quadre dopo il comando `\caption`.

```
\caption[Corta]{Luuuuuunnnngaaaaaaa}
```

Con `\label` e `\ref`, si possono creare riferimenti a oggetti mobili nel testo.

L'esempio seguente disegna un quadrato e lo inserisce nel documento. Lo si può usare per riservare spazio per illustrazioni che verrano inserite a documento ultimato.

```
La figura~\ref{white} \'e un esempio di pop-art.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Cinque per cinque centimetri.} \label{white}
\end{figure}
```

In questo esempio, LaTeX cercherà *disperatamente* (`!`) di mettere la figura *qui* (`h`)[13]. Se questo non è possibile, prova a posizionare la figura *in fondo* alla pagina (`b`). Non potendo mettere la figura nella pagina corrente, stabilisce se è possibile creare una pagina di oggetti mobili contenente questa figura ed

[13] Nell'ipotesi che la coda delle figure sia vuota.

eventuali tabelle dalla coda delle tabelle. Se non c'è abbastanza materiale per una pagina di questo tipo, LaTeX inizia una nuova pagina, e ancora una volta tratta la figura come se fosse appena comparsa nel testo.

In alcune circostanze potrebbe essere necessario usare i comandi

`\clearpage` o `\cleardoublepage`.

Essi ordinano a LaTeX di stampare immediatamente tutti gli oggetti mobili rimasti nelle code e cominciare una nuova pagina. `\cleardoublepage` va ad una nuova pagina dispari.

Più avanti in questa guida è spiegato come si fa a includere immagini PostScript all'interno di documenti LaTeX 2_ε.

Capitolo 3

Scrivere formule matematiche

Ora si può cominciare! In questo capitolo verrà esplorato il principale punto di forza del TEX: la scrittura di formule matematiche. Un avvertimento: questo capitolo intacca solamente la superficie dell'argomento. Benché le cose che verranno spiegate siano sufficienti per la maggior parte delle persone, non bisogna perdere la speranza se non si trova la soluzione per 'quella' particolare necessità di scrittura matematica. È molto probabile che il problema sia risolvibile con l'$\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX[1] o con qualche altro pacchetto.

3.1 Informazioni generali

Il LATEX ha una modalità speciale per scrivere la matematica. In un paragrafo le parti matematiche si inseriscono tra `\(` e `\)`, tra `$` e `$` o tra `\begin{math}` e `\end{math}`.

```
Somma $a$ al quadrato e $b$ al
quadrato per avere $c$ al quadrato,
o, per usare un approccio pi\`u
matematico: $c^{2}=a^{2}+b^{2}$
```

Somma a al quadrato e b al quadrato per avere c al quadrato, o, per usare un approccio più matematico: $c^2 = a^2 + b^2$

```
\TeX{} va pronunciato
$\tau\epsilon\chi$.\\[6pt]
100~m$^{3}$ d'acqua\\[6pt]
Questo arriva dal mio $\heartsuit$
```

TEX va pronunciato $\tau\epsilon\chi$.

100 m^3 d'acqua

Questo arriva dal mio $\heartsuit$

È preferibile *evidenziare* grosse equazioni matematiche o formule, piuttosto che scriverle come testo normale su linee separate. Questo significa

[1] CTAN:/tex-archive/macros/latex/required/amslatex

includerle tra `\[` e `\]` o tra `\begin{displaymath}` e `\end{displaymath}`. A questa maniera si ottengono formule non numerate. Se si vuole che il LaTeX le numeri, occorre usare l'ambiente `equation`.

```
Somma $a$ al quadrato e $b$ al
quadrato per avere $c$ al quadrato,
o, per usare un approccio pi\`u
matematico:
\begin{displaymath}
c^{2}=a^{2}+b^{2}
\end{displaymath}
E un'altra riga tanto per gradire.
```

Somma a al quadrato e b al quadrato per avere c al quadrato, o, per usare un approccio più matematico:

$$c^{2}=a^{2}+b^{2}$$

E un'altra riga tanto per gradire.

Si può avere un riferimento a un'equazione tramite `\label` e `\ref`:

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
Dall'equazione (\ref{eq:eps})
si deduce che \ldots
```

$$\epsilon > 0 \tag{3.1}$$

Dall'equazione (3.1) si deduce che …

Occorre notare che le equazioni sono scritte in due maniere differenti a seconda che siano all'interno di un paragrafo o meno:

```
$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$
```

$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

Ci sono delle differenze tra il *modo matematico* e il *modo testo*; per esempio nel *modo matematico*:

1. Quasi tutti gli spazi e le interruzioni di linea non hanno significato, poicheé gli spazi o sono derivati dalla struttura dell'espressione matematica o, se occorre, devono essere specificati usando comandi speciali come `\,`, `\quad` o `\qquad`.

2. Non sono ammesse linee vuote. Una sola formula per paragrafo.

3. Ogni lettera è considerata come un nome di variabile e viene scritta come tale. Se si vuole avere del testo normale all'interno di una formula (col normale *font* 'diritto' e spaziatura normale), allora bisogna scriverlo usando il comando `\textrm{...}`.

```
\begin{equation}
\forall x \in \mathbf{R}:
\qquad x^{2} \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R}: \qquad x^{2} \geq 0 \tag{3.2}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\textrm{per ogni }x\in\mathbf{R}
\end{equation}
```

$$x^{2} \geq 0\qquad \textrm{per ogni }x\in\mathbf{R} \tag{3.3}$$

I matematici possono essere molto puntigliosi riguardo ai simboli da usare: nell'esempio precedente sarebbe stato più convenzionale usare il 'blackboard bold', che si ottiene con `\mathbb`, disponibile tramite il pacchetto amsfonts o amssymb. L'ultimo esempio diventerebbe

```
\begin{displaymath}
x^{2} \geq 0\qquad
\textrm{per ogni }x\in\mathbb{R}
\end{displaymath}
```

$$x^{2} \geq 0\qquad \textrm{per ogni }x\in\mathbb{R}$$

3.2 Raggruppamenti in modo matematico

La maggior parte dei comandi agiscono solo sul carattere successivo. Se si vuole che un comando abbia effetto su più caratteri, si deve raggrupparli usando le parentesi graffe: `{...}`.

```
\begin{equation}
a^x+y \neq a^{x+y}
\end{equation}
```

$$a^x+y \neq a^{x+y} \tag{3.4}$$

3.3 Elementi base di una formula matematica

In questa sezione verranno descritti i comandi più importanti che vengono usati per scrivere formule matematiche. Si dia un'occhiata alla sezione 3.10 alla pagina 52 per una lista dettagliata dei comandi per scrivere simboli matematici.

Le **lettere greche minuscole** si ottengono con `\alpha`, `\beta`, `\gamma`, ..., quelle maiuscole con `\Gamma`, `\Delta`, ...[2]

[2]Non c'è l'Alpha maiuscolo in LaTeX 2_ε perché è uguale alla normale A romana. Quando il nuovo encoding matematico sarà finito le cose cambieranno.

```
$\lambda,\xi,\pi,\mu,\Phi,\Omega$
```

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

Esponenti e deponenti possono essere scritti usando i caratteri `^` e `_`.

```
$a_{1}$ \qquad $x^{2}$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$\\
$e^{x^2} \neq {e^x}^2$
```

$a_1 \qquad x^2 \qquad e^{-\alpha t} \qquad a^3_{ij}$
$e^{x^2} \neq {e^x}^2$

La **radice quadrata** si scrive come `\sqrt`, la radice n-esima si ottiene con `\sqrt[`n`]`. La dimensione della radice è determinata automaticamente dal LaTeX. Se si vuole solo il simbolo, si può usare `\surd`.

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$\\[3pt]
$\surd[x^2 + y^2]$
```

$\sqrt{x} \qquad \sqrt{x^2+\sqrt{y}} \qquad \sqrt[3]{2}$
$\surd[x^2 + y^2]$

I comandi `\overline` e `\underline` creano delle **linee orizzontali** direttamente sopra o sotto un'espressione.

```
$\overline{m+n}$
```

$\overline{m+n}$

I comandi `\overbrace` e `\underbrace` creano delle lunghe **graffe orizzontali** sopra o sotto un'espressione.

```
$\underbrace{ a+b+\cdots+z }_{26}$
```

$\underbrace{a+b+\cdots+z}_{26}$

Per aggiungere un accento matematico, come una piccola freccia o un segno di tilde alle variabili si possono usare i comandi della tabella 3.1 a pagina 52. I segni di cappello o tilde coprenti diversi caratteri sono prodotti tramite `\widetilde` e `\widehat`. Il simbolo `'` produce un segno di 'primo'.

```
\begin{displaymath}
y=x^{2}\qquad y'=2x\qquad y''=2
\end{displaymath}
```

$$y = x^2 \qquad y' = 2x \qquad y'' = 2$$

I **vettori** spesso sono indicati aggiungendo un piccolo simbolo freccia sopra una variabile. Questo viene fatto tramite il comando `\vec`. I due comandi `\overrightarrow` e `\overleftarrow` sono utili per indicare il vettore da A a B.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

$$\vec a \quad \overrightarrow{AB}$$

I nomi delle funzioni, come 'log', sono spesso stampati in *font* dritto piuttosto che in corsivo come le variabili, perciò il LaTeX fornisce i seguenti comandi per i principali nomi di funzione:

```
\arccos  \cos   \csc  \exp  \ker     \limsup  \min  \sinh
\arcsin  \cosh  \deg  \gcd  \lg      \ln      \Pr   \sup
\arctan  \cot   \det  \hom  \lim     \log     \sec  \tan
\arg     \coth  \dim  \inf  \liminf  \max     \sin  \tanh
```

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}=1$$

Per quel che riguarda la funzione modulo, ci sono due comandi: `\bmod` per l'operatore binario "$a \bmod b$" e `\pmod` per espressioni come "$x \equiv a \pmod{b}$".

Una **frazione** si ottiene con il comando `\frac{...}{...}`. A volte la forma 1/2 è preferibile, perché è più bella a vedersi per piccole quantità di 'materiale frazionario'.

```
1 ora e~$\frac{1}{2}$
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

1 ora e $\frac{1}{2}$

$$\frac{ x^{2} }{ k+1 }\qquad x^{ \frac{2}{k+1} }\qquad x^{ 1/2 }$$

Per scrivere coefficienti binomiali o strutture simili, si può usare alternativamente `{... \choose ...}` o `{... \atop ...}`. Il secondo comando produce lo stesso risultato del primo, ma senza parentesi (occorre notare come l'uso di questi comandi 'vecchio stile' sia espressamente vietato dal pacchetto amsmath. Essi sono rimpiazzati da `\binom` e `\genfrac`. Il secondo è una forma generale da cui si possono derivare tutti i costrutti correlati, per esempio si può ottenere un comando similare ad `\atop` con
`\newcommand{\newatop}[2]{\genfrac{}{}{0pt}{1}{#1}{#2}}`).

```
\begin{displaymath}
{n \choose k}\qquad {x \atop y+2}
\end{displaymath}
```

$$\binom{n}{k}\qquad \genfrac{}{}{0pt}{}{x}{y+2}$$

Per le relazioni binarie può essere utile mettere un simbolo sopra un altro. `\stackrel` mette il simbolo indicato nel primo argomento sopra quello specificato nel secondo argomento, scrivendo il primo con dimensioni inferiori (pari a quelle di un apice) mentre il secondo viene mantenuto nella sua posizione usuale.

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

L'**operatore integrale** è generato tramite `\int`, l'**operatore sommatoria** con `\sum` e l'**operatore produttoria** con `\prod`. I limiti superiore e inferiore sono specificati con `^` e `_`, proprio come gli esponenti e i deponenti[3].

```
\begin{displaymath}
\sum_{i=1}^{n} \qquad
\int_{0}^{\frac{\pi}{2}} \qquad
\prod_\epsilon
\end{displaymath}
```

$$\sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad \prod_\epsilon$$

Per ciò che riguarda le **parentesi** e gli altri **delimitatori**, ci sono tutti i tipi possibili di simboli nel TEX (es. $[\ \langle\ \|\ \updownarrow$). Le parentesi tonde e quadre possono essere scritte 'normalmente', ossia con i caratteri corrispondenti, mentre le parentesi graffe vanno scritte con `\{`; tutti gli altri delimitatori vengono generati tramite comandi speciali (es. `\updownarrow`). Per una lista di tutti i delimitatori disponibili si veda la tabella 3.8 alla pagina 54.

```
\begin{displaymath}
{a,b,c}\neq\{a,b,c\}
\end{displaymath}
```

$$a, b, c \neq \{a, b, c\}$$

Se si mette il comando `\left` davanti ad un operatore di apertura o `\right` davanti ad un operatore di chiusura, il TEX determinerà automaticamente la giusta dimensione del delimitatore. Si noti che occorre chiudere ogni `\left` con un corrispondente `\right`, e che la dimensione viene correttamente determinata solo se tutti e due sono sulla stessa riga. Se non si vuole niente sulla destra, si può usare l'invisibile '`\right.`'!

```
\begin{displaymath}
1 + \left( \frac{1}{ 1-x^{2} }
    \right) ^3
\end{displaymath}
```

$$1 + \left(\frac{1}{1-x^{2}} \right)^3$$

In certi casi è necessario specificare manualmente la giusta dimensione di un delimitatore matematico, e ciò si può fare tramite i comandi `\big`, `\Big`, `\bigg` e `\Bigg`, utilizzabili come prefissi sulla maggior parte dei delimitatori[4].

[3] AMS-LATEX, inoltre, permette esponenti e deponenti multilinea.

[4] Questi comandi non funzionano come ci si aspetterebbe se si è usato un comando per

```
$\Big( (x+1) (x-1) \Big) ^{2}$\\
$\big(\Big(\bigg(\Bigg($\quad
$\big\}\Big\}\bigg\}\Bigg\}$\quad
$\big\|\Big\|\bigg\|\Bigg\|$
```

$\Big((x+1)(x-1) \Big)^{2}$
$\big(\Big(\bigg(\Bigg(\quad \big\}\Big\}\bigg\}\Bigg\}\quad \big\|\Big\|\bigg\|\Bigg\|$

Per inserire **tre puntini** in una formula si possono usare diversi comandi. `\ldots` inserisce i puntini sulla linea di base del testo, `\cdots` li centra rispetto alla riga. Oltre a ciò ci sono i comandi `\vdots` per i puntini verticali e `\ddots` per i puntini diagonali. Si possono trovare altri esempi nella sezione 3.5.

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

3.4 Spazi in modo matematico

Se gli spazi all'interno delle formule scelti da TEX non sono soddisfacenti, possono essere modificati inserendo appositi comandi di spaziatura. Ci sono alcuni comandi per gli spazi piccoli: `\,` per $\frac{3}{18}$ quad (␣), `\:` per $\frac{4}{18}$ quad (␣) e `\;` per $\frac{5}{18}$ quad (␣). La sequenza di escape-spazio `\␣` genera uno spazio di dimensione media mentre `\quad` (␣) e `\qquad` (␣) producono spazi ampi. La dimensione di un `\quad` corrisponde all'ampiezza del carattere 'M' nel *font* corrente. Il comando `\!` produce uno spazio negativo di $-\frac{3}{18}$ quad (␣).

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_{D} g(x,y)
  \, \ud x\, \ud y
\end{displaymath}
invece di
\begin{displaymath}
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}
```

$$\int\!\!\!\int_{D} g(x,y)\, \mathrm{d} x\, \mathrm{d} y$$

invece di

$$\int\int_{D} g(x,y)\mathrm{d}x\mathrm{d}y$$

Si noti che la 'd' nel differenziale viene solitamente scritta in *roman*.

il cambio della dimensione del *font* o se si è specificato l'opzione `11pt` o `12pt`. Si può usare il pacchetto `exscale` o `amsmath` per correggere questo comportamento.

$\mathcal{AMS}$-LaTeX fornisce altri metodi per regolare lo spazio tra segni di integrazione multipli, e cioè i comandi `\iint`, `\iiint`, `\iiiint` e `\idotsint`. Se è stato caricato il pacchetto **amsmath**, l'esempio precedente può essere scritto in questo modo:

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \, \ud x \, \ud y
\end{displaymath}
```

$$\iint_{D} \, \mathrm{d} x \, \mathrm{d} y$$

Si veda il documento elettronico `testmath.tex` (distribuito con $\mathcal{AMS}$-LaTeX) oppure il capitolo 8 di "The LaTeX Companion" per ulteriori dettagli.

3.5 Materiale allineato verticalmente

Per scrivere **matrici** si usa l'ambiente `array`. Funziona in modo simile all'ambiente `tabular`. Il comando `\\` viene usato per interrompere le righe.

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

L'ambiente `array` può anche essere usato per scrivere espressioni che hanno un solo grande delimitatore usando un "`.`" come delimitatore destro (`\right`) invisibile:

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 a & \textrm{se $d>c$}\\
 b+x & \textrm{di mattina}\\
 l & \textrm{per tutto il giorno}
  \end{array} \right.
\end{displaymath}
```

$$y = \left\{ \begin{array}{ll} a & \textrm{se $d>c$}\\ b+x & \textrm{di mattina}\\ l & \textrm{per tutto il giorno} \end{array} \right.$$

Come nell'ambiente `tabular` anche nell'ambiente `array` si possono disegnare linee, ad esempio per separare gli elementi di una matrice:

```
\begin{displaymath}
\left(\begin{array}{c|c}
 1 & 2 \\
\hline
3 & 4
\end{array}\right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}\right)$$

Per formule che occupano diverse righe o per sistemi di equazioni si possono usare gli ambienti `eqnarray` e `eqnarray*` al posto di `equation`. In `eqnarray` ad ogni riga viene attribuito un numero. In `eqnarray*` non viene numerato nulla.

Gli ambienti `eqnarray` e `eqnarray*` funzionano come una tabella a 3 colonne nella forma `{rcl}`, dove la colonna centrale può essere usata per il segno di uguaglianza o disuguaglianza. O per qualunque altro segno si pensi di usare. Il comando `\\` interrompe le righe.

```
\begin{eqnarray}
f(x) & = & \cos x     \\
f'(x) & = & -\sin x   \\
\int_{0}^{x} f(y)dy &
 = & \sin x
\end{eqnarray}
```

$$
\begin{aligned}
f(x) &= \cos x && (3.5)\\
f'(x) &= -\sin x && (3.6)\\
\int_{0}^{x} f(y)dy &= \sin x && (3.7)
\end{aligned}
$$

Si noti che lo spazio da entrambi i lati del segno di uguaglianza è piuttosto grande. Può essere ridotto impostando `\setlength\arraycolsep{2pt}`, come nell'esempio seguente.

Le **equazioni lunghe** non vengono automaticamente divise in parti "pulite". L'autore deve specificare dove interromperle e di quanto farle rientrare. I due metodi seguenti sono i più comunemente usati per questo scopo.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x -\frac{x^{3}}{3!}
     +\frac{x^{5}}{5!}-{}
                        \nonumber\\
 & & {}-\frac{x^{7}}{7!}+{}\cdots
\end{eqnarray}}
```

$$
\begin{aligned}
\sin x = {} & x-\frac{x^{3}}{3!}+\frac{x^{5}}{5!}-{}\\
& {}-\frac{x^{7}}{7!}+{}\cdots && (3.8)
\end{aligned}
$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1
     -\frac{x^{2}}{2!} +{} }
                        \nonumber\\
 & & {}+\frac{x^{4}}{4!}
     -\frac{x^{6}}{6!}+{}\cdots
\end{eqnarray}
```

$$
\begin{aligned}
&\cos x = 1-\frac{x^{2}}{2!}+{}\\
&\qquad\qquad {}+\frac{x^{4}}{4!}-\frac{x^{6}}{6!}+{}\cdots && (3.9)
\end{aligned}
$$

Il comando `\nonumber` comunica a LaTeX di non generare un numero per l'equazione.

Può essere difficoltoso ottenere delle equazioni ben allineate verticalmente; il pacchetto **amsmath** fornisce un insieme di alternative più potenti (si vedano gli ambienti `split` e `align`).

3.6 Fantasmi

I fantasmi non si vedono, però occupano dello spazio nella mente di molte persone. LaTeX non è diverso. Possiamo usare i fantasmi per alcuni trucchi interessanti sulla spaziatura.

A volte, quando si allinea verticalmente del testo usando `^` e `_`, LaTeX è un po' troppo di aiuto. Usando il comando `\phantom` si può riservare dello spazio per caratteri che non compariranno nell'output finale. È bene osservare l'esempio seguente.

```
\begin{displaymath}
{}^{12}_{\phantom{1}6}\textrm{C}
\qquad \textrm{invece di} \qquad
{}^{12}_{6}\textrm{C}
\end{displaymath}
```

$${}^{12}_{6}\textrm{C} \qquad \textrm{invece di} \qquad {}^{12}_{6}\textrm{C}$$

```
\begin{displaymath}
\Gamma_{ij}^{\phantom{ij}k}
\qquad \textrm{invece di} \qquad
\Gamma_{ij}^{k}
\end{displaymath}
```

$$\Gamma_{ij}^{k} \qquad \textrm{invece di} \qquad \Gamma_{ij}^{k}$$

3.7 Dimensione dei *font* matematici

In modo matematico TeX seleziona la dimensione del *font* in accordo col contesto. Gli apici, per esempio, vengono scritti in un *font* più piccolo. Se si vuole scrivere una parte di una formula in *roman* non si deve usare il comando `\textrm`, perché il meccanismo di cambio di dimensione non funziona, dato che `\textrm` temporaneamente entra in modo testo. Si usa invece `\mathrm` per mantere attivo il meccanismo di cambio di dimensione. Ma si faccia attenzione, `\mathrm` funzionerà bene solo su brevi parti. Gli spazi non sono ancora attivi e i caratteri accentati non funzionano[5].

```
\begin{equation}
2^{\textrm{nd}} \quad
2^{\mathrm{nd}}
\end{equation}
```

$$2^{\textrm{nd}} \quad 2^{\mathrm{nd}} \tag{3.10}$$

Nonostante tutto a volte è necessario comunicare a LaTeX la corretta dimensione del *font*. In modo matematico la dimensione del *font* viene impostata con quattro comandi:

[5] Il pacchetto AMS-LaTeX fa funzionare il comando `\textrm` con il cambiamento di dimensione, quindi in realtà nell'esempio riportato non si vede alcuna differenza tra i due comandi; si provi a compilare l'esempio senza l'ausilio dell'AMS-LaTeX per apprezzare la differenza. [N.d.T.]

`\displaystyle` (123), `\textstyle` (123), `\scriptstyle` ($\scriptstyle 123$) e `\scriptscriptstyle` ($\scriptscriptstyle 123$).

Il cambiamento di stile influisce anche sul modo in cui vengono scritti i limiti.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
 \frac{\displaystyle
   \sum_{i=1}^n(x_i-\overline x)
   (y_i-\overline y)}
  {\displaystyle\biggl[
 \sum_{i=1}^n(x_i-\overline x)^2
\sum_{i=1}^n(y_i-\overline y)^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\mathop{\mathrm{corr}}(X,Y)=\frac{\displaystyle\sum_{i=1}^n(x_i-\overline x)(y_i-\overline y)}{\displaystyle\biggl[\sum_{i=1}^n(x_i-\overline x)^2\sum_{i=1}^n(y_i-\overline y)^2\biggr]^{1/2}}$$

Questo è uno di quegli esempi in cui si ha bisogno di parentesi più larghe di quanto non forniscano i comandi `\left[` `\right]`.

3.8 Teoremi, leggi, ...

Componendo documenti matematici, sicuramente è necessario disporre di un metodo per formattare "Lemmi", "Definizioni", "Assiomi" e strutture simili; il LaTeX fornisce ciò con il comando

`\newtheorem{`*nome*`}[`*contatore*`]{`*testo*`}[`*sezione*`]`

L'argomento *nome* è una breve parola chiave utilizzata per identificare successivamente il "teorema". Con l'argomento *testo* invece viene indicato il nome del "teorema", che poi verrà stampato sul documento.

Gli argomenti tra parentesi quadre sono opzionali; entrambi vengono utilizzati per specificare la numerazione utilizzata per il "teorema". L'argomento *contatore* deve specificare il *nome* di un "teorema" precedentemente dichiarato, il nuovo "teorema" verrà allora numerato con la stessa sequenza numerica. L'argomento *sezione* permette di specificare a quale tipo di unità di sezionamento (*chapter, section, ...*) collegare la numerazione del "teorema".

Dopo aver eseguito il comando `\newtheorem` nel preambolo del documento, è possibile utilizzare il seguente comando al suo interno.

> `\begin{`*nome*`}[`*testo*`]`
> Questo è il mio interessantissimo teorema
> `\end{`*nome*`}`

La teoria dovrebbe essere abbastanza, i seguenti esempi dovrebbero rimuovere i dubbi finali e rendere chiaro come l'ambiente `\newtheorem` sia obbiettivamente troppo complesso da capire.

```
% Definizioni per il documento
% Preambolo
\newtheorem{legge}{Legge}
\newtheorem{decrt}[legge]{Decreto}
% Corpo del documento
\begin{legge} \label{legge:box}
Il capo ha ragione
\end{legge}
\begin{decrt}[Aggiornamento alla
Legge~\ref{legge:box}]
Il capo ha \emph{sempre} ragione
\end{decrt}
\begin{legge}
Se il capo ha torto, vedere la
legge~\ref{legge:box}
\end{legge}
```

Legge 1 *Il capo ha ragione*

Decreto 2 (Aggiornamento alla Legge 1) *Il capo ha* sempre *ragione*

Legge 3 *Se il capo ha torto, vedere la legge 1*

Il teorema "Decreto" utilizza lo stesso contatore del teorema "Legge", perciò avrà un numero di identificazione che segue la stessa sequenza numerica utilizzata da questo (effettivamente il numero di identificazione è il successore di quello dell'altro). L'argomento tra parentesi quadre viene utilizzato per specificare il nome del teorema (o qualcosa di affine).

```
\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
Se esistono due o pi\`u modi
per fare una cosa, e se uno
di questi modi pu\`o creare
una catastrofe, allora
qualcuno lo sceglier\`a.\end{mur}
```

Murphy 3.8.1 *Se esistono due o più modi per fare una cosa, e se uno di questi modi può creare una catastrofe, allora qualcuno lo sceglierà.*

Il teorema "Murphy" ha il numero di identificazione che è collegato al numero del paragrafo corrente (*section*). È possibile specificare anche un'altra unità in luogo del paragrafo, come ad esempio *chapter* (capitolo) o *subsection* (sottoparagrafo).

3.9 Simboli in grassetto

È abbastanza difficile ottenere simboli grassettati in LaTeX; probabilmente questo è intenzionale dato che compositori amatoriali spesso tendono ad abusarne. Il comando per il cambio del *font* `\mathbf`, restituisce lettere grassettate, che tuttavia sono in roman, quando usualmente i simboli matematici sono in corsivo. È presente anche il comando `\boldmath`, il quale tuttavia *può essere utilizzato solo all'esterno dell'ambiente matematico*, comunque questo funziona anche per i simboli.

```
\begin{displaymath}
\mu, M \qquad \mathbf{M} \qquad
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```

$$\mu, M \qquad \mathbf{M} \qquad \boldsymbol{\mu, M}$$

Si noti che anche la virgola è grassettata, anche se ciò non è richiesto.

Il pacchetto amsby (incluso da amsmath) rende tutto questo molto più semplice, dato che include il comando `\boldsymbol`.

```
\begin{displaymath}
\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \qquad \boldsymbol{\mu}, \boldsymbol{M}$$

3.10 Lista dei simboli matematici

Nelle seguenti tabelle si possono trovare tutti i simboli ai quali si può accedere normalmente dall'*ambiente matematico*.

Per utilizzare i simboli elencati nelle Tabelle 3.12–3.16[6] il pacchetto amssymb deve essere caricato nel preambolo del documento ed inoltre i *font* $\mathcal{AMS}$ math devono essere installati sul sistema. Se il pacchetto $\mathcal{AMS}$ ed i *font* non sono installati, controllare presso `CTAN:/tex-archive/macros/latex/required/amslatex`

Tabella 3.1: Accenti in modalità matematica.

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

Tabella 3.2: Lettere greche minuscole.

α	`\alpha`	θ	`\theta`	o	`o`	υ	`\upsilon`
β	`\beta`	ϑ	`\vartheta`	π	`\pi`	ϕ	`\phi`
γ	`\gamma`	ι	`\iota`	ϖ	`\varpi`	φ	`\varphi`
δ	`\delta`	κ	`\kappa`	ρ	`\rho`	χ	`\chi`
ϵ	`\epsilon`	λ	`\lambda`	ϱ	`\varrho`	ψ	`\psi`
ε	`\varepsilon`	μ	`\mu`	σ	`\sigma`	ω	`\omega`
ζ	`\zeta`	ν	`\nu`	ς	`\varsigma`		
η	`\eta`	ξ	`\xi`	τ	`\tau`		

Tabella 3.3: Lettere greche maiuscole.

Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

[6] Queste tabelle derivano dal `symbols.tex` di David Carlisle e sono state cambiate in modo esauriente come suggerito da Josef Tkadlec.

Tabella 3.4: Relazioni binarie.

È possibile produrre le corrispondenti negazioni aggiungendo un comando `\not` come prefisso ai seguenti simboli.

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` o `\le`	$\geq$	`\geq` o `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$	`\sqsubset` [a]	$\sqsupset$	`\sqsupset` [a]	$\Join$	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` oppure `\ne`

[a] Utilizzare il pacchetto latexsym per accedere a questo simbolo

Tabella 3.5: Operatori binari.

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$\ast$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lor`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]	$\wr$	`\wr`

Tabella 3.6: Operatori BIG.

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`	$\bigoplus$	`\bigoplus`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`			$\bigodot$	`\bigodot`
$\int$	`\int`	$\oint$	`\oint`			$\biguplus$	`\biguplus`

Tabella 3.7: Frecce.

$\leftarrow$	`\leftarrow` o `\gets`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\rightarrow$	`\rightarrow` o `\to`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\iff$	`\iff` (con spazi grandi)	$\leadsto$	`\leadsto` [a]

[a]Utilizzare il pacchetto latexsym per accedere a questo simbolo

Tabella 3.8: Delimitatori.

$($	`(`	$)$	`)`	$\uparrow$	`\uparrow`	$\Uparrow$	`\Uparrow`
$[$	`[` o `\lbrack`	$]$	`]` o `\rbrack`	$\downarrow$	`\downarrow`	$\Downarrow$	`\Downarrow`
$\{$	`\{` o `\lbrace`	$\}$	`\}` o `\rbrace`	$\updownarrow$	`\updownarrow`	$\Updownarrow$	`\Updownarrow`
$\langle$	`\langle`	$\rangle$	`\rangle`	$\vert$	`|` o `\vert`	$\Vert$	`\\|` o `\Vert`
$\lfloor$	`\lfloor`	$\rfloor$	`\rfloor`	$\lceil$	`\lceil`	$\rceil$	`\rceil`
$/$	`/`	$\backslash$	`\backslash`		. (carattere fantasma)		

Tabella 3.9: Grandi delimitatori.

$\lgroup$	`\lgroup`	$\rgroup$	`\rgroup`	$\lmoustache$	`\lmoustache`	$\rmoustache$	`\rmoustache`
$\arrowvert$	`\arrowvert`	$\Arrowvert$	`\Arrowvert`	$\bracevert$	`\bracevert`		

Tabella 3.10: Simboli misti.

$\dots$	`\dots`	$\cdots$	`\cdots`	$\vdots$	`\vdots`	$\ddots$	`\ddots`
$\hbar$	`\hbar`	$\imath$	`\imath`	$\jmath$	`\jmath`	ℓ	`\ell`
$\Re$	`\Re`	$\Im$	`\Im`	$\aleph$	`\aleph`	$\wp$	`\wp`
$\forall$	`\forall`	$\exists$	`\exists`	$\mho$	`\mho` [a]	∂	`\partial`
$'$	`'`	$\prime$	`\prime`	$\emptyset$	`\emptyset`	∞	`\infty`
∇	`\nabla`	$\triangle$	`\triangle`	$\Box$	`\Box` [a]	$\Diamond$	`\Diamond` [a]
$\bot$	`\bot`	$\top$	`\top`	$\angle$	`\angle`	$\surd$	`\surd`
$\diamondsuit$	`\diamondsuit`	$\heartsuit$	`\heartsuit`	$\clubsuit$	`\clubsuit`	$\spadesuit$	`\spadesuit`
$\neg$	`\neg` o `\lnot`	$\flat$	`\flat`	$\natural$	`\natural`	$\sharp$	`\sharp`

[a] Utilizzare il pacchetto latexsym per accedere a questo simbolo

Tabella 3.11: Simboli non matematici.

Questi simboli possono venire utilizzati anche in modo testo.

†	`\dag`	§	`\S`	©	`\copyright`
‡	`\ddag`	¶	`\P`	£	`\pounds`

Tabella 3.12: Delimitatori $\mathcal{AMS}$.

$\ulcorner$	`\ulcorner`	$\urcorner$	`\urcorner`	$\llcorner$	`\llcorner`	$\lrcorner$	`\lrcorner`

Tabella 3.13: $\mathcal{AMS}$ Greco ed Ebraico.

$\digamma$	`\digamma`	$\varkappa$	`\varkappa`	$\beth$	`\beth`	$\daleth$	`\daleth`	$\gimel$	`\gimel`

Tabella 3.14: Relazioni binarie $\mathcal{AMS}$.

$\lessdot$	`\lessdot`	$\gtrdot$	`\gtrdot`	$\doteqdot$	`\doteqdot` o `\Doteq`
$\leqslant$	`\leqslant`	$\geqslant$	`\geqslant`	$\risingdotseq$	`\risingdotseq`
$\eqslantless$	`\eqslantless`	$\eqslantgtr$	`\eqslantgtr`	$\fallingdotseq$	`\fallingdotseq`
$\leqq$	`\leqq`	$\geqq$	`\geqq`	$\eqcirc$	`\eqcirc`
$\lll$	`\lll` o `\llless`	$\ggg$	`\ggg` o `\gggtr`	$\circeq$	`\circeq`
$\lesssim$	`\lesssim`	$\gtrsim$	`\gtrsim`	$\triangleq$	`\triangleq`
$\lessapprox$	`\lessapprox`	$\gtrapprox$	`\gtrapprox`	$\bumpeq$	`\bumpeq`
$\lessgtr$	`\lessgtr`	$\gtrless$	`\gtrless`	$\Bumpeq$	`\Bumpeq`
$\lesseqgtr$	`\lesseqgtr`	$\gtreqless$	`\gtreqless`	$\thicksim$	`\thicksim`
$\lesseqqgtr$	`\lesseqqgtr`	$\gtreqqless$	`\gtreqqless`	$\thickapprox$	`\thickapprox`
$\preccurlyeq$	`\preccurlyeq`	$\succcurlyeq$	`\succcurlyeq`	$\approxeq$	`\approxeq`
$\curlyeqprec$	`\curlyeqprec`	$\curlyeqsucc$	`\curlyeqsucc`	$\backsim$	`\backsim`
$\precsim$	`\precsim`	$\succsim$	`\succsim`	$\backsimeq$	`\backsimeq`
$\precapprox$	`\precapprox`	$\succapprox$	`\succapprox`	$\vDash$	`\vDash`
$\subseteqq$	`\subseteqq`	$\supseteqq$	`\supseteqq`	$\Vdash$	`\Vdash`
$\Subset$	`\Subset`	$\Supset$	`\Supset`	$\Vvdash$	`\Vvdash`
$\sqsubset$	`\sqsubset`	$\sqsupset$	`\sqsupset`	$\backepsilon$	`\backepsilon`
$\therefore$	`\therefore`	$\because$	`\because`	$\varpropto$	`\varpropto`
$\shortmid$	`\shortmid`	$\shortparallel$	`\shortparallel`	$\between$	`\between`
$\smallsmile$	`\smallsmile`	$\smallfrown$	`\smallfrown`	$\pitchfork$	`\pitchfork`
$\vartriangleleft$	`\vartriangleleft`	$\vartriangleright$	`\vartriangleright`	$\blacktriangleleft$	`\blacktriangleleft`
$\trianglelefteq$	`\trianglelefteq`	$\trianglerighteq$	`\trianglerighteq`	$\blacktriangleright$	`\blacktriangleright`

Tabella 3.15: Frecce $\mathcal{AMS}$.

$\dashleftarrow$	`\dashleftarrow`	$\dashrightarrow$	`\dashrightarrow`	$\multimap$	`\multimap`
$\leftleftarrows$	`\leftleftarrows`	$\rightrightarrows$	`\rightrightarrows`	$\upuparrows$	`\upuparrows`
$\leftrightarrows$	`\leftrightarrows`	$\rightleftarrows$	`\rightleftarrows`	$\downdownarrows$	`\downdownarrows`
$\Lleftarrow$	`\Lleftarrow`	$\Rrightarrow$	`\Rrightarrow`	$\upharpoonleft$	`\upharpoonleft`
$\twoheadleftarrow$	`\twoheadleftarrow`	$\twoheadrightarrow$	`\twoheadrightarrow`	$\upharpoonright$	`\upharpoonright`
$\leftarrowtail$	`\leftarrowtail`	$\rightarrowtail$	`\rightarrowtail`	$\downharpoonleft$	`\downharpoonleft`
$\leftrightharpoons$	`\leftrightharpoons`	$\rightleftharpoons$	`\rightleftharpoons`	$\downharpoonright$	`\downharpoonright`
$\Lsh$	`\Lsh`	$\Rsh$	`\Rsh`	$\rightsquigarrow$	`\rightsquigarrow`
$\looparrowleft$	`\looparrowleft`	$\looparrowright$	`\looparrowright`	$\leftrightsquigarrow$	`\leftrightsquigarrow`
$\curvearrowleft$	`\curvearrowleft`	$\curvearrowright$	`\curvearrowright`		
$\circlearrowleft$	`\circlearrowleft`	$\circlearrowright$	`\circlearrowright`		

Tabella 3.16: Relazioni binarie e frecce negate $\mathcal{AMS}$.

$\nless$	`\nless`	$\ngtr$	`\ngtr`	$\varsubsetneqq$	`\varsubsetneqq`
$\lneq$	`\lneq`	$\gneq$	`\gneq`	$\varsupsetneqq$	`\varsupsetneqq`
$\nleq$	`\nleq`	$\ngeq$	`\ngeq`	$\nsubseteqq$	`\nsubseteqq`
$\nleqslant$	`\nleqslant`	$\ngeqslant$	`\ngeqslant`	$\nsupseteqq$	`\nsupseteqq`
$\lneqq$	`\lneqq`	$\gneqq$	`\gneqq`	$\nmid$	`\nmid`
$\lvertneqq$	`\lvertneqq`	$\gvertneqq$	`\gvertneqq`	$\nparallel$	`\nparallel`
$\nleqq$	`\nleqq`	$\ngeqq$	`\ngeqq`	$\nshortmid$	`\nshortmid`
$\lnsim$	`\lnsim`	$\gnsim$	`\gnsim`	$\nshortparallel$	`\nshortparallel`
$\lnapprox$	`\lnapprox`	$\gnapprox$	`\gnapprox`	$\nsim$	`\nsim`
$\nprec$	`\nprec`	$\nsucc$	`\nsucc`	$\ncong$	`\ncong`
$\npreceq$	`\npreceq`	$\nsucceq$	`\nsucceq`	$\nvdash$	`\nvdash`
$\precneqq$	`\precneqq`	$\succneqq$	`\succneqq`	$\nvDash$	`\nvDash`
$\precnsim$	`\precnsim`	$\succnsim$	`\succnsim`	$\nVdash$	`\nVdash`
$\precnapprox$	`\precnapprox`	$\succnapprox$	`\succnapprox`	$\nVDash$	`\nVDash`
$\subsetneq$	`\subsetneq`	$\supsetneq$	`\supsetneq`	$\ntriangleleft$	`\ntriangleleft`
$\varsubsetneq$	`\varsubsetneq`	$\varsupsetneq$	`\varsupsetneq`	$\ntriangleright$	`\ntriangleright`
$\nsubseteq$	`\nsubseteq`	$\nsupseteq$	`\nsupseteq`	$\ntrianglelefteq$	`\ntrianglelefteq`
$\subsetneqq$	`\subsetneqq`	$\supsetneqq$	`\supsetneqq`	$\ntrianglerighteq$	`\ntrianglerighteq`
$\nleftarrow$	`\nleftarrow`	$\nrightarrow$	`\nrightarrow`	$\nleftrightarrow$	`\nleftrightarrow`
$\nLeftarrow$	`\nLeftarrow`	$\nRightarrow$	`\nRightarrow`	$\nLeftrightarrow$	`\nLeftrightarrow`

Tabella 3.17: Operatori binari $\mathcal{AMS}$.

$\dotplus$	`\dotplus`	$\centerdot$	`\centerdot`	$\intercal$	`\intercal`
$\ltimes$	`\ltimes`	$\rtimes$	`\rtimes`	$\divideontimes$	`\divideontimes`
$\Cup$	`\Cup` o `\doublecup`	$\Cap$	`\Cap` o `\doublecap`	$\smallsetminus$	`\smallsetminus`
$\veebar$	`\veebar`	$\barwedge$	`\barwedge`	$\doublebarwedge$	`\doublebarwedge`
$\boxplus$	`\boxplus`	$\boxminus$	`\boxminus`	$\circleddash$	`\circleddash`
$\boxtimes$	`\boxtimes`	$\boxdot$	`\boxdot`	$\circledcirc$	`\circledcirc`
$\leftthreetimes$	`\leftthreetimes`	$\rightthreetimes$	`\rightthreetimes`	$\circledast$	`\circledast`
$\curlyvee$	`\curlyvee`	$\curlywedge$	`\curlywedge`		

Tabella 3.18: $\mathcal{AMS}$ misti.

$\hbar$	\hbar	$\hslash$	\hslash	$\Bbbk$	\Bbbk
$\square$	\square	$\blacksquare$	\blacksquare	$\circledS$	\circledS
$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle	$\complement$	\complement
$\triangledown$	\triangledown	$\blacktriangledown$	\blacktriangledown	$\Game$	\Game
$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge	$\bigstar$	\bigstar
$\angle$	\angle	$\measuredangle$	\measuredangle	$\sphericalangle$	\sphericalangle
$\diagup$	\diagup	$\diagdown$	\diagdown	$\backprime$	\backprime
$\nexists$	\nexists	$\Finv$	\Finv	$\varnothing$	\varnothing
$\eth$	\eth	$\mho$	\mho		

Tabella 3.19: Grafie Matematiche.

Esempio	Comando	Pacchetto richiesto
ABCdef	\mathrm{ABCdef}	
ABCdef	\mathit{ABCdef}	
$\mathnormal{ABCdef}$	\mathnormal{ABCdef}	
$\mathcal{ABC}$	\mathcal{ABC}	
$\mathscr{ABC}$	\mathscr{ABC}	mathrsfs
$\mathcal{ABC}$	\mathcal{ABC}	eucal con opzione: mathcal o
	\mathscr{ABC}	eucal con opzione: mathscr
$\mathfrak{ABCdef}$	\mathfrak{ABCdef}	eufrak
$\mathbb{ABC}$	\mathbb{ABC}	amsfonts oppure amssymb

Capitolo 4

Specialità

Nel preparare documenti di una certa dimensione, LaTeX fornisce un valido aiuto in virtù di alcune sue speciali caratteristiche quali la possibilità di creare l'indice analitico, la gestione dei riferimenti bibliografici, ed altro ancora. Una più completa descrizione delle particolarità e delle potenzialità di LaTeX la si può trovare nel *LaTeX Manual* [1] e in *The LaTeX Companion* [3].

4.1 Inserimento di grafici EPS

LaTeX, per mezzo degli ambienti `figure` e `table`, mette a disposizione alcune agevolazioni essenziali per gestire oggetti mobili come le immagini o i disegni.

Vi sono anche varie possibilità per creare dei veri grafici con LaTeX puro oppure con un pacchetto aggiuntivo di LaTeX. Sfortunatamente, la maggior parte degli utenti le considera del tutto incomprensibili. Per questo motivo, questa possibilità non sarà più considerata per il resto del manuale. Per ulteriori informazioni sull'argomento si possono consultare *The LaTeX Companion* [3] e il *LaTeX Manual* [1].

Un metodo di gran lunga più semplice per inserire dei grafici in un documento consiste nel prepararli attraverso un *software* specifico[1] e di includere il grafico ottenuto nel documento. Anche in questo caso, i pacchetti di LaTeX offrono molte soluzioni al riguardo. In questa introduzione verrà unicamente discusso l'utilizzo di grafici Encapsulated PostScript (EPS) perché veramente semplice ed ampiamente diffuso. Al fine di utilizzare immagini in formato EPS è necessario possedere una stampante PostScript[2].

Un buon insieme di comandi per inserire i grafici è fornito dal pacchetto

[1] Ad esempio: XFig, CorelDraw!, Freehand, Gnuplot,...

[2] Un'altra possibilità per gestire il formato PostScript consiste nell'utilizzare il programma GhostScript disponibile presso il sito `CTAN:/tex-archive/support/ghostscript`. Gli utenti *Windows* potrebbero considerare, per lo stesso scopo, il programma GSview.

graphicx di D. P. Carlisle; fa parte di una intera famiglia di estensioni nota come gruppo dei *pacchetti grafici* ("*graphics*" *bundle*)[3].

In un sistema dotato di una stampante PostScript e avente il pacchetto graphicx installato, la sequenza di istruzioni per includere un'immagine in un documento è la seguente:

1. Dal programma grafico utilizzato, si salvi l'immagine in formato EPS[4].

2. Si inserisca, nel preambolo del file sorgente, il comando per caricare il pacchetto graphicx

   `\usepackage[`*driver*`]{graphicx}`

   dove *driver* è il proprio programma di conversione "da dvi in PostScript". Quello di gran lunga più diffuso è `dvips`. Il nome del *driver* è necessario in quanto non esiste in TEX un metodo standard per includere i grafici. Conoscendo il nome del *driver*, il pacchetto graphicx è in grado di scegliere il sistema corretto per inserire l'informazione sui grafici nel file `.dvi` affinché la stampante sia in grado di riconoscerla e possa inglobare, in maniera corretta, il file `.eps`.

3. Si utilizzi il comando

   `\includegraphics[`*opzione*=*valore*,...`]{`*file*`}`

   per inserire il *file* nel documento. Il parametro facoltativo permette l'inserimento di una lista, separata da virgole, di *opzioni* con relativi *valori*. Le *opzioni* possono essere usate per modificare la larghezza, l'altezza e l'orientamento del grafico incluso. La tabella 4.1 elenca le più importanti opzioni.

L'esempio che segue dovrebbe rendere più chiaro il procedimento:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test}
\end{center}
\end{figure}
```

[3] `CTAN:/tex-archive/macros/latex/required/graphics`

[4] Se il programma grafico non prevede il formato EPS si può provare ad installare un *driver* per stampanti PostScript (ad esempio, per le LaserWriter della Apple) e quindi stampare su *file* utilizzando questo *driver*. Con una buona dose di fortuna, questo file sarà in formato EPS. Si osservi che un file EPS non deve contenere più di una pagina. Alcuni *driver* per stampanti possono essere esplicitamente configurati per produrre il formato EPS.

Tabella 4.1: Opzioni per il pacchetto graphicx.

`width`	ridimensiona il grafico alla larghezza specificata
`height`	ridimensiona il grafico all'altezza specificata
`angle`	ruota il grafico in senso orario
`scale`	riassegna le dimensioni del grafico

In questo modo viene incluso il grafico contenuto nel file `test.eps`. Il grafico viene *prima* ruotato di 90 gradi e *poi* ne viene variata la larghezza fino a raggiungere la metà della larghezza di un normale paragrafo. Il rapporto di scala resta 1,0 dal momento che non viene specificata alcuna altezza particolare. La larghezza e l'altezza possono essere assegnate anche in dimensioni assolute. Per maggiori informazioni, si consulti la tabella 5.5 di pagina 76. Se si desidera approfondire l'argomento, sarà utile leggere [8] e [11].

4.2 La bibliografia

Per realizzare una bibliografia si ricorre all'ambiente `thebibliography`. Ciascuna voce viene inserita con il comando

`\bibitem{`*nome-simbolico*`}`

Il *nome-simbolico* è poi utilizzato per indicare il libro, l'articolo o la pubblicazione all'interno del documento mediante il comando `\cite`.

`\cite{`*nome-simbolico*`}`

La numerazione dei riferimenti bibliografici è generata automaticamente. Il parametro che segue il comando `\begin{thebibliography}` assegna l'ampiezza massima di questa numerazione. Nel seguente esempio, `{99}` informa LaTeX di aspettarsi che non vi sarà alcun riferimento bibliografico con un numero d'ordine superiore a 99.

```
Partl~\cite{pa} ha
suggerito che \dots

\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Vol.~9, No.~1 ('88)
\end{thebibliography}
```

Partl [1] ha suggerito che ...

Bibliografia

[1] H. Partl: *German TEX*, TUGboat Vol. 9, No. 1 ('88)

Per progetti più ambiziosi si potrebbe aver bisogno di utilizzare il programma BibTEX. BibTEX è presente nella maggior parte delle distribuzioni TEX. Consente di gestire un database di riferimenti bibliografici e di estrarne quelli relativi alle citazioni presenti nel documento. L'aspetto visibile delle bibliografie generate da BibTEX si fonda sul concetto dei fogli di stile per mezzo dei quali è possibile creare dei riferimenti bibliografici a partire da una vasta gamma di modelli prestabiliti.

4.3 L'indice analitico

L'indice analitico costituisce, in molti libri, una caratteristica di grande utilità. Con LATEX e il programma di supporto `makeindex`[5], un indice analitico viene generato in modo veramente semplice. In questa introduzione verranno illustrati solo i comandi essenziali per generare l'indice analitico. Per un'analisi più approfondita, si consulti *The LATEX Companion* [3].

Per abilitare la composizione dell'indice analitico da parte di LATEX, si deve richiamare nel preambolo il pacchetto makeidx per mezzo del comando:

```
\usepackage{makeidx}
```

mentre i comandi relativi alla creazione dell'indice devono essere attivati dal comando

```
\makeindex
```

inserito sempre nel preambolo del file sorgente.

[5] Su sistemi che non permettono l'utilizzo, per i file, di nomi più lunghi degli 8 caratteri, il nome del programma potrebbe essere `makeidx`.

Tabella 4.2: Esempi di voci dell'indice analitico.

Esempio	**Voce**	**Commento**
`\index{hello}`	hello, 1	Voce primaria
`\index{hello!Peter}`	Peter, 3	Voce secondaria sotto 'hello'
`\index{Sam@\textsl{Sam}}`	*Sam*, 2	Voce con forma specificata
`\index{Lin@\textbf{Lin}}`	**Lin**, 7	Come sopra
`\index{Jenny|textbf}`	Jenny, **3**	Pagina con forma specificata
`\index{Joe|textit}`	Joe, *5*	Come sopra

Il contenuto dell'indice analitico viene specificato attraverso il comando

`\index{`*argomento*`}`

dove *argomento* indica la voce dell'indice analitico; tale comando va inserito nel punto del testo dove si desidera che punti la voce presente nell'indice analitico. La tabella 4.2 illustra, per mezzo di alcuni esempi, la sintassi del parametro *argomento*.

Quando il programma LaTeX viene eseguito sul file sorgente, ogni comando `\index` scrive un'opportuna voce insieme al numero di pagina corrente in un file speciale che ha il medesimo nome del file sorgente LaTeX, ma estensione diversa (`.idx`). Questo file `.idx` viene poi elaborato dal programma `makeindex`.

`makeindex` *nome-del-file*

Il programma `makeindex` genera un indice ordinato in un file avente ancora lo stesso nome, ma questa volta estensione `.ind`. Se a questo punto si compila ancora una volta il sorgente LaTeX, tale indice analitico ordinato verrà incluso nel documento nel punto dove LaTeX trova il comando

`\printindex`

Il pacchetto showidx distribuito con LaTeX 2_ε stampa le voci dell'indice analitico nel margine sinistro della pagina. Ciò risulta essere di grande utilità in fase di correzione di bozze e controllo dell'indice analitico.

4.4 Intestazioni eleganti

Il pacchetto fancyhdr[6], scritto da Piet van Oostrum, fornisce pochi e semplici comandi attraverso i quali è possibile personalizzare l'intestazione e il piè di pagina di un documento. Un possibile esempio di utilizzo di questo pacchetto lo si ha osservando la parte in alto di questa pagina.

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% i comandi seguenti impediscono la scrittura in maiuscolo
% dei nomi dei capitoli e dei paragrafi nelle intestazioni
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}  % rimuove l'attuale contenuto dell'intestazione
            % e del pi\`e di pagina
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % riserva spazio per la linea
\fancypagestyle{plain}{%
   \fancyhead{} % ignora, nello stile plain, le intestazioni
   \renewcommand{\headrulewidth}{0pt} % e la linea
}
```

Figura 4.1: Esempio di utilizzo di fancyhdr.

La reale difficoltà nel personalizzare le intestazioni e i piè di pagina consiste nel dovervi inserire cose come i nomi della sezione e del capitolo corrente. LaTeX la risolve con un approccio a due tappe. Nella definizione dell'intestazione e del piè di pagina si adoperano i comandi `\rightmark` e `\leftmark` per indicare rispettivamente l'intestazione del capitolo e quella della sezione in corso. I valori di questi due comandi vengono aggiornati ogni volta che viene elaborato un comando di inizio capitolo o sezione.

Al fine di garantire la massima flessibilità, i comandi `\chapter` e soci non vanno a modificare direttamente `\rightmark` e `\leftmark`, bensì richiamano un ulteriore comando, ossia `\chaptermark`, `\sectionmark` oppure `\subsectionmark`, che è il vero responsabile della nuova definizione di `\rightmark` e `\leftmark`.

[6] Presso il `CTAN:/tex-archive/macros/latex/contrib/supported/fancyhdr`.

Così se si desidera cambiare l'aspetto del nome del capitolo nell'intestazione si deve semplicemente "riassegnare" il comando `\chaptermark`.

La figura 4.1 illustra un esempio di utilizzo del pacchetto fancyhdr in base al quale si realizza un'intestazione simile a quella di questo libretto. Ad ogni modo conviene procurarsi la documentazione relativa al pacchetto dall'indirizzo richiamato nella nota a piè di pagina.

4.5 Il pacchetto Verbatim

In precedenza è stato introdotto, in questo libro, l'*ambiente* `verbatim`. In questa sezione studieremo il *pacchetto* verbatim. Il pacchetto verbatim è essenzialmente una riscrittura dell'ambiente `verbatim` al fine di superare alcuni dei limiti dell'ambiente `verbatim` originale. Va da sé che non vi è nulla di spettacolare, però con la creazione del pacchetto verbatim sono state inserite nuove funzionalità ed è per questa ragione che se ne fa menzione qui. Il pacchetto verbatim mette a disposizione il comando

`\verbatiminput{`*nome-del-file*`}`

che permette di inserire del testo ASCII grezzo in un documento come se fosse all'interno di un ambiente `verbatim`.

Poiché il pacchetto verbatim fa parte di quel gruppo di pacchetti noti come '*tools*', lo si trova già installato sulla maggior parte dei sistemi. Per saperne di più su questo pacchetto si legga l'articolo [9].

4.6 La protezione dei comandi fragili

Il testo passato come argomento in comandi quali `\caption` o `\section` può comparire più di una volta all'interno di un documento (ad esempio nell'indice del libro così come nel corpo del documento). Alcuni comandi falliscono se passati come argomento di comandi tipo `\section` e, pertanto, si dicono comandi fragili. Sono fragili, ad esempio, comandi come `\footnote` oppure `\phantom`. Quello di cui, questi comandi fragili, hanno bisogno per funzionare è la protezione (non è vero anche per tutti noi?). È possibile proteggerli facendoli precedere dal comando `\protect`.

`\protect` fa riferimento al solo comando che segue subito a destra ma non ai suoi argomenti. Nella maggior parte dei casi un superfluo `\protect` non farà danni.

```
\section{Io sono premuroso
        \protect\footnote{e proteggo le mie note}}
```

Capitolo 5

Personalizzare il LaTeX

I documenti prodotti usando i comandi appresi finora saranno considerati soddisfacenti da una larga fascia di persone. Sebbene non abbiano un aspetto fantasioso, obbediscono alle leggi della buona tipografia, cosa che li rende facili da leggere e piacevoli da guardare.

Ci sono tuttavia situazioni in cui il LaTeX non fornisce comandi o ambienti adatti alle intenzioni dell'autore, o in cui il risultato di un comando esistente non ne soddisfa le esigenze.

In questo capitolo verranno esposti alcuni suggerimenti su come insegnare al LaTeX nuovi trucchi, e su come fare in modo che produca risultati diversi da quelli predefiniti.

5.1 Comandi, ambienti e pacchetti nuovi

Si nota subito che tutti i comandi introdotti in questo libro sono scritti in un riquadro, e che sono poi elencati in un indice in fondo al libro. Invece di usare direttamente i comandi LaTeX necessari, è stato creato un pacchetto in cui sono stati definiti nuovi comandi ed ambienti per questo scopo; così, si può scrivere semplicemente:

```
\begin{command}
\ci{dum}
\end{command}
```

`\dum`

In questo esempio è stato usato un nuovo ambiente, chiamato `command`, che disegna il riquadro attorno al comando, ed un nuovo comando, chiamato `\ci`, che scrive il nome del comando ed inserisce la voce corrispondente nell'indice. Ciò può essere verificato andando a cercare il comando `\dum` nell'indice in fondo al libro, dove si può trovare la voce `\dum`, con riferimenti ad ogni pagina dove il comando `\dum` è stato citato.

Se in un qualunque momento si decidesse di non volere più un riquadro attorno ai comandi, si può semplicemente cambiare la definizione dell'ambiente `command`, per dargli un nuovo aspetto; ciò è molto più semplice di dover andare a sostituire i comandi usati per disegnare il riquadro in ogni singolo posto in cui sono stati usati.

5.1.1 Nuovi comandi

Per aggiungere comandi personali, si usa il comando

`\newcommand{`*nome*`}[`*num*`]{`*definizione*`}`

Il comando richiede due argomenti: il *nome* del nuovo comando, e la *definizione* del comando. L'opzione *num* in parentesi quadre è opzionale, e specifica il numero di argomenti che il nuovo comando richiede (fino ad un massimo di 9); se manca, si assume il valore predefinito 0, ossia nessun argomento.

I due esempi seguenti dovrebbero rendere l'idea. Il primo esempio definisce un nuovo comando, chiamato `\umtb`, che è l'abbreviazione di "Una (mica tanto) breve introduzione al LaTeX 2_ε"; un comando simile potrebbe essere utile se si dovesse scrivere ripetutamente il titolo di questo libro.

```
\newcommand{\umtb}{Una (mica tanto)
    breve introduzione al
    \LaTeXe}
Questa \'e ``\umtb''
    \ldots{} ``\umtb''
```

Questa è "Una (mica tanto) breve introduzione al LaTeX 2_ε" ... "Una (mica tanto) breve introduzione al LaTeX 2_ε"

Il prossimo esempio mostra come definire un comando che accetta argomenti. Il parametro `#1` viene sostituito dal primo argomento specificato; nel caso di più argomenti, questi diventano `#2`, `#3`, ... e così via.

```
\newcommand{\txsit}[1]
 {Questa \'e una \emph{#1} breve
     introduzione al \LaTeXe}
% nel corpo del documento:
\begin{itemize}
\item \txsit{(mica tanto)}
\item \txsit{molto}
\end{itemize}
```

- Questa è una *(mica tanto)* breve introduzione al LaTeX 2_ε
- Questa è una *molto* breve introduzione al LaTeX 2_ε

Il LaTeX non permette di creare nuovi comandi che sovrascrivano quelli già esistenti; c'è però un comando speciale, da usare nel caso in cui ciò si voglia esplicitamente fare: `\renewcommand`; la sintassi è la stessa di `\newcommand`.

In certi casi, conviene usare il comando `\providecommand`: funziona come `\newcommand`, ma viene silenziosamente ignorato dal LaTeX 2_ε se il comando definito esiste già.

Ci sono alcune osservazioni da fare sugli spazi seguenti un comando LaTeX. Per maggiori informazioni, vedere pagina 6.

5.1.2 Nuovi ambienti

Il comando `\newenvironment`, corrispondente di `\newcommand`, permette di creare ambienti personali. La sua sintassi è la seguente:

`\newenvironment{`*nome*`}[`*num*`]{`*prima*`}{`*dopo*`}`

Come per `\newcommand`, il comando `\newenvironment` si può usare con o senza un argomento opzionale. Ciò che viene specificato nell'argomento *prima* viene elaborato prima del testo dell'ambiente; ciò che viene specificato nell'argomento *dopo* viene elaborato quando viene incontrato il comando `\end{`*nome*`}`.

Il seguente esempio mostra l'uso del comando `\newenvironment`:

```
\newenvironment{regio}
 {\rule{1ex}{1ex}%
      \hspace{\stretch{1}}}
 {\hspace{\stretch{1}}%
      \rule{1ex}{1ex}}

\begin{regio}
Miei umili sudditi\ldots
\end{regio}
```

■ Miei umili sudditi… ■

L'argomento *num* si usa come nel comando `\newcommand`. Il LaTeX si assicura che non venga definito un ambiente già esistente: se si vuole cambiare un ambiente già definito, si usa il comando `\renewenvironment`, che ha la stessa sintassi di `\newenvironment`.

I comandi usati in questo esempio verranno spiegati più avanti: il comando `\rule` a pagina 81, `\stretch` a pagina 75, `\hspace` a pagina 75.

5.1.3 Un pacchetto personale

Quando vengono definiti molti nuovi ambienti e comandi, il preambolo del documento può diventare piuttosto lungo; in questo caso, è una buona idea creare un pacchetto LaTeX contenente le definizioni dei propri comandi ed ambienti; si può poi usare il comando `\usepackage` per rendere il pacchetto disponibile nel documento.

```
% Pacchetto dimostrativo di Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\lntc}{Una (mica tanto) breve introduzione al \LaTeXe}
\newcommand{\txsit}[1]{Una \emph{#1} breve
                        introduzione al \LaTeXe}
\newenvironment{regio}{\begin{quote}}{\end{quote}}
```

Figura 5.1: Esempio di pacchetto

Essenzialmente, per scrivere un pacchetto si copia il contenuto del preambolo del documento in un file a parte, con un nome terminante per `.sty`. C'è poi il comando speciale

`\ProvidesPackage{`*nome pacchetto*`}`

da usare all'inizio del file contenente pacchetto. `\ProvidesPackage` dice al LaTeX il nome del pacchetto, permettendogli di emettere messaggi di errore appropriati quando si tenta di includere due volte lo stesso pacchetto. Nella figura 5.1 è mostrato un piccolo pacchetto di esempio, contenente i comandi definiti precedentemente.

5.2 Caratteri

5.2.1 Comandi per cambiare il carattere

Il LaTeX sceglie il carattere appropriato, con l'opportuna dimensione, in base alla struttura logica del documento (sezioni, note a piè di pagina, ...). In alcuni casi, si potrebbe però voler cambiare il carattere ed il corpo, manualmente; per far ciò, si possono usare i comandi elencati nelle tabelle 5.1 e 5.2. Il corpo effettivo di ogni carattere è una questione di design, e dipende dalla classe del documento e dalle sue opzioni; nella tabella 5.3 sono elencati i corpi in punti per questi comandi, configurate dalle classi standard.

```
{\small I piccoli e %
\textbf{coraggiosi}
Romani governavano}
{\Large tutta la grande
e grossa \textit{Italia}.}
```

I piccoli e **coraggiosi** Romani governavano tutta la grande e grossa *Italia*.

Un'importante caratteristica del LaTeX 2_ε è che gli attributi dei *font* sono indipendenti: è cioè possibile specificare comandi per cambiare le dimensioni e persino il carattere, mantenendo gli attributi di grassetto o corsivo specificati prima.

In modalità matematica (*math mode*) si possono usare i comandi di cambiamento dei caratteri per uscire temporaneamente dal *math mode* ed inserire testo normale. Se si vuole usare un carattere diverso per la matematica, esistono comandi specifici, elencati nella tabella 5.4.

Insieme ai comandi di specifica dei caratteri, le parentesi graffe hanno un ruolo importante: vengono usate per costruire *gruppi*; i gruppi limitano il raggio d'azione della maggior parte dei comandi LaTeX.

```
Gli piacciono {\LARGE grandi
e {\small piccole}, le lettere}.
```

Gli piacciono grandi e piccole, le lettere.

I comandi per il cambiamento della dimensione cambiano anche l'interlinea, ma solo se il paragrafo termina entro il raggio d'azione del comando; di conseguenza, la parentesi graffa chiusa non dovrebbe essere messa troppo presto: osservate la posizione del comando `\par` nei due seguenti esempi.[1]

[1] Il comando `\par` è equivalente a una linea vuota

Tabella 5.1: Caratteri (tono e direzione)

`\textrm{...}`	romano	`\textsf{...}`	sans serif
`\texttt{...}`	`typewriter`		
`\textmd{...}`	normale	`\textbf{...}`	**grassetto**
`\textup{...}`	tondo	`\textit{...}`	*corsivo*
`\textsl{...}`	*inclinato*	`\textsc{...}`	MAIUSCOLETTO
`\emph{...}`	*enfatizzato*	`\textnormal{...}`	*font* del documento

Tabella 5.2: Dimensioni

`\tiny`	*font* minuscolo	`\Large`	*font* più grande
`\scriptsize`	*font* molto piccolo	`\LARGE`	*font* molto grande
`\footnotesize`	*font* abbastanza piccolo	`\huge`	enorme
`\small`	*font* piccolo	`\Huge`	il più grande
`\normalsize`	*font* normale		
`\large`	*font* grande		

Tabella 5.3: Dimensioni assolute in punti (corpi) nelle classi standard

corpo	10pt (predefinito)	opzione 11pt	opzione 12pt
`\tiny`	5pt	6pt	6pt
`\scriptsize`	7pt	8pt	8pt
`\footnotesize`	8pt	9pt	10pt
`\small`	9pt	10pt	11pt
`\normalsize`	10pt	11pt	12pt
`\large`	12pt	12pt	14pt
`\Large`	14pt	14pt	17pt
`\LARGE`	17pt	17pt	20pt
`\huge`	20pt	20pt	25pt
`\Huge`	25pt	25pt	25pt

Tabella 5.4: Caratteri matematici.

Comando	*Esempio*	*Risultato*
`\mathcal{...}`	`$\mathcal{B}=c$`	$\mathcal{B}=c$
`\mathrm{...}`	`$\mathrm{K}_2$`	K_2
`\mathbf{...}`	`$\sum x=\mathbf{v}$`	$\sum x=\mathbf{v}$
`\mathsf{...}`	`$\mathsf{G\times R}$`	$\mathsf{G\times R}$
`\mathtt{...}`	`$\mathtt{L}(b,c)$`	$\mathtt{L}(b,c)$
`\mathnormal{...}`	`$\mathnormal{R_{19}}\neq R_{19}$`	$\mathnormal{R_{19}}\neq R_{19}$
`\mathit{...}`	`$\mathit{ffi}\neq ffi$`	$\mathit{ffi}\neq ffi$

```
{\Large Non leggere questo!
Non \'e vero! Puoi credermi!\par}
```

Non leggere questo! Non è vero! Puoi credermi!

```
{\Large Neanche questo \'e vero.
Ma ri\-cor\-da che sono un
bugiardo.}\par
```

Neanche questo è vero. Ma ricorda che sono un bugiardo.

Se si desidera cambiare la dimensione del carattere per un intero paragrafo, o più, conviene usare la versione ambiente dei comandi di cambiamento:

```
\begin{Large}
Questo non \'e vero. Ma d'altra
parte, cosa lo \'e, di questi
tempi\ldots
\end{Large}
```

Questo non è vero. Ma d'altra parte, cosa lo è, di questi tempi…

Questo vi salverà dal dover contare innumerevoli parentesi graffe.

5.2.2 Pericolo, Will Robinson, Pericolo

Come osservato all'inizio di questo capitolo, è pericoloso ingombrare il documento con comandi espliciti come quelli elencati finora, perché lavorano in contrasto alla filosofia del LaTeX, che consiste nel separare l'aspetto logico e visivo del documento. Questo significa che se si utilizza lo stesso tipo di carattere in più punti, sempre per lo stesso tipo di informazione, è opportuno usare `\newcommand` per definire un "guscio logico" per il comando di cambiamento del carattere.

```
\newcommand{\oops}[1]{\textbf{#1}}
Non \oops{entrare} in questa
stanza, \'e abitata da una
\oops{macchina} la cui origine ed
il cui scopo sono sconosciuti.
```

Non **entrare** in questa stanza, è abitata da una **macchina** la cui origine ed il cui scopo sono sconosciuti.

Questo modo di fare ha il vantaggio che, se in seguito si decide di usare un'altra rappresentazione visiva del pericolo, lo si può fare senza cercare nel documento tutte le istanze di `\textbf` per poi vedere se è stato usato per il pericolo o per qualche altra ragione.

5.2.3 Consiglio

Per concludere il nostro viaggio nella terra dei caratteri e dei corpi, ecco un piccolo consiglio:

> **<u>Ricorda!</u>** *tanti* **Pi** ù*font* scegli di usare **in** un `documento`, *tanto* più LEGGIBILE e ***bello*** *diventa*.

5.3 Spaziatura

5.3.1 Interlinea

Se si vuole cambiare l'interlinea di un documento, si può usare il comando

`\linespread{`*fattore*`}`

nel preambolo; l'interlinea "uno e mezzo" si ottiene con `\linespread{1.3}`, interlinea due con `\linespread{1.6}`. L'interlinea normale si ottiene con un fattore 1.

5.3.2 Formattazione dei paragrafi

Il LaTeX ha due parametri che influenzano l'aspetto dei paragrafi, che possono essere modificati con istruzioni del tipo

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}.
```

Lo scopo di queste istruzioni è di aumentare lo spazio tra paragrafi e di ridurre a zero il rientro iniziale, come richiesto dalle regole tipografiche di alcuni paesi dell'Europa continentale; bisogna però stare attenti all'effetto che queste istruzioni hanno sull'indice: anche le sue voci vengono maggiormente spaziate; per evitare questo effetto, si possono spostare le due istruzioni precedenti da qualche parte dopo il comando `\tableofcontents`, oppure li si può evitare del tutto, poiché la maggior parte dei libri professionali usano un rientro iniziale ed una spaziatura normale fra paragrafi.

Se si vuole rientrare un paragrafo che non verrebbe normalmente rientrato, si può usare il comando

`\indent`

all'inizio del paragrafo[2]. Ovviamente, ciò avrà un qualche effetto solo se `\parindent` non è impostata a zero.

[2] Per attivare il rientro del primo paragrafo dopo ogni titolo di sezione, si può usare il pacchetto indentfirst del gruppo 'tools'.

Per creare un paragrafo senza rientro, si usa il comando

```
\noindent
```

all'inizio del paragrafo; questo può tornare utile quando si voglia iniziare un documento direttamente con il testo, e non con un comando di sezione.

5.3.3 Spazio orizzontale

Il LaTeX determina automaticamente lo spazio tra parole e frasi; se si vuole aggiungere spazio orizzontale, si usa:

```
\hspace{lungh}
```

Qualora lo spazio debba essere mantenuto anche quando capita alla fine o all'inizio di una riga, si usa il comando `\hspace*` invece di `\hspace`. *lungh*, nel caso più semplice, è solo un numero seguito da un'unita di misura (le principali unità di misura sono elencate nella tabella 5.5.)

```
Questo\hspace{1.5cm}\'e uno
spazio di 1,5 cm.
```

Questo   è uno spazio di 1,5 cm.

Il comando

```
\stretch{n}
```

genera uno speciale spazio elastico, che si allunga finché tutto lo spazio rimanente sulla riga non viene eliminato; se due comandi `\hspace{\stretch{n}}` vengono usati sulla stella linea, si allungano proporzionalmente al fattore di elasticità.

```
x\hspace{\stretch{1}}
x\hspace{\stretch{3}}x
```

x   x   x

5.3.4 Spazio verticale

Lo spazio tra paragrafi, sezioni, sottosezioni, ... viene determinato automaticamente dal LaTeX; se necessario, si può aggiungere più spazio *tra due paragrafi* con il comando:

```
\vspace{lungh}
```

Normalmente, questo comando va usato tra due righe vuote; se lo spazio

Tabella 5.5: Unità di misura del TeX.

`mm`	millimetro $\approx 1/25$ pollici	$\sqcup$
`cm`	centimetro = 10 mm	$\sqcup$
`in`	pollice (*inch*) = 25,4 mm	$\sqcup$
`pt`	punto $\approx 1/72$ pollici $\approx \frac{1}{3}$ mm	$\sqcup$
`em`	(circa) larghezza di 'M' nel *font* attuale	$\sqcup$
`ex`	(circa) altezza di 'x' nel *font* attuale	$\sqcup$

deve essere conservato in cima o in fondo alla pagina, è bene usare la versione con asterisco del comando: `\vspace*` invece di `\vspace`.

Il comando `\stretch`, usato insieme a `\pagebreak`, permette di scrivere testo sull'ultima riga di una pagina, o centrato verticalmente sulla pagina.

```
Un po' di testo\ldots

\vspace{\stretch{1}}
Questo va in fondo alla pagina.\pagebreak
```

Spazio aggiuntivo tra due linee dello *stesso* paragrafo (o tabella) viene specificato da

`\\[`*lungh*`].`

5.4 Layout di pagina

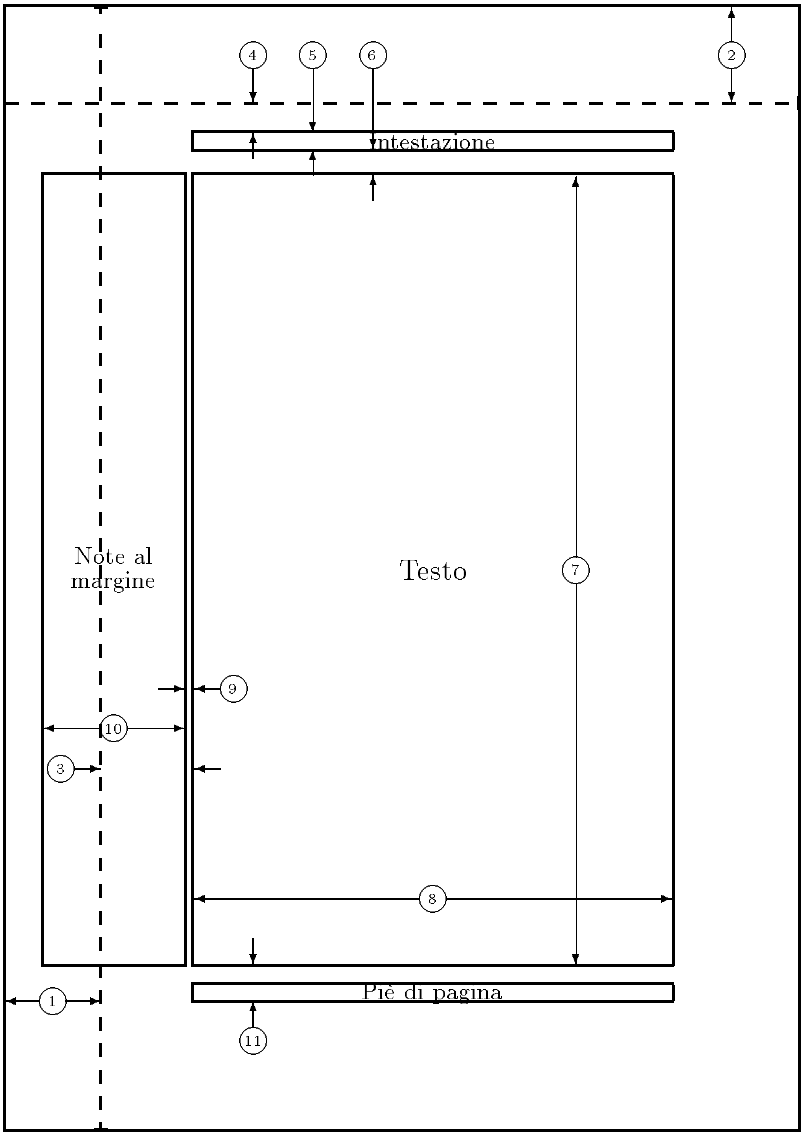
Il LaTeX 2_ε permette di specificare le dimensioni del foglio nel comando `\documentclass`, e sceglie poi di conseguenza i margini giusti; certe volte, tuttavia, si potrebbe non essere soddisfatti dei valori predefiniti, e naturalmente questi si possono cambiare. Nella figura 5.2 sono indicati tutti i parametri che possono essere cambiati. La figura è stata prodotta usando il pacchetto layout del gruppo 'tools'[3].

CALMA!. . . prima di buttarsi nella frenesia dell' "Allarghiamo un po' questa strettissima pagina" è bene riflettere un attimo. Come per la maggior parte delle cose in LaTeX, c'è una buona ragione per cui il layout della pagina è quello che è.

Certo, paragonato ad una pagina creata da *MS Word*, sembra mostruosamente stretta; ma basta dare un'occhiata ad uno qualsiasi dei propri libri[4]

[3] `CTAN:/tex-archive/macros/latex/required/tools`

[4] Libri veri, stampati da case editrici serie.



1	un pollice + \hoffset	2	un pollice + \voffset
3	\evensidemargin = 70pt	4	\topmargin = 22pt
5	\headheight = 13pt	6	\headsep = 19pt
7	\textheight = 595pt	8	\textwidth = 360pt
9	\marginparsep = 7pt	10	\marginparwidth = 106pt
11	\footskip = 27pt		\marginparpush = 5pt (non mostrato)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 597pt		\paperheight = 845pt

Figura 5.2: Parametri del layout di pagina.

preferiti e contare il numero di caratteri di una linea di testo normale: si scopre che non ci sono più di circa 66 caratteri per riga; ora, facendo lo stesso conteggio su una pagina prodotta dal LaTeX, si scopre che ci sono pure circa 66 caratteri per riga. L'esperienza mostra che leggere diventa più difficile appena ci sono più caratteri per riga: questo dipende dal fatto che l'occhio si affatica spostandosi dalla fine di una riga all'inizio della successiva; questo è anche il motivo per cui i giornali vengono stampati su più colonne.

Quindi, quando si aumenta la larghezza del testo, bisogna tenere in mente che si rende difficile la vita del lettore. Ma basta con queste cautele, poiché ho promesso di dire come fare...

Il LaTeX ha due comandi per modificare questi parametri, che vengono usati, normalmente, nel preambolo del documento.

Il primo comando assegna un determinato valore ad un qualsiasi parametro:

`\setlength{`*parametro*`}{`*lungh*`}`

Il secondo comando aggiunge una certa lunghezza ad un qualsiasi parametro:

`\addtolength{`*parametro*`}{`*lungth*`}`

Questo secondo comando è in genere più utile di `\setlength`, poiché permette di lavorare relativamente alle impostazioni preesistenti. Per esempio, per aggiungere un centimetro alla larghezza complessiva del testo, si usa questo comando nel preambolo del documento:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

In questo contesto potrebbe essere utile dare un'occhiata al pacchetto calc, che permette operazioni aritmetiche con gli argomenti di `\setlength`, ed in altri posti dove si possono inserire valori numerici come argomenti di funzioni.

5.5 Divertirsi con le lunghezze

Quando possibile, è meglio cercare di evitare lunghezze assolute, nei documenti LaTeX; è meglio basarsi sulla larghezza o altezza di qualche elemento della pagina; per esempio, la larghezza di una figura può essere eguagliata a `\textwidth` per riempire completamente la pagina.

I tre seguenti comandi permettono di determinare larghezza, altezza e profondità di una stringa di testo:

\settoheight{*comando*}{*testo*} \settodepth{*comando*}{*testo*} \settowidth{*comando*}{*testo*}

Il seguente esempio mostra come si possono applicare i precedenti comandi:

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[0pt][r]{#1:\ }}{}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Dove}$a$,
$b$ -- sono adiacenti all'angolo
retto di un triangolo rettangolo.

$c$ -- \`e l'ipotenusa del
triangolo e si sente sola.

$d$ -- infine, non si vede proprio,
qui. Non \`e stupefacente?
\end{vardesc}
```

$$a^2 + b^2 = c^2$$

Dove: a, b – sono adiacenti all'angolo retto di un triangolo rettangolo.

c – è l'ipotenusa del triangolo e si sente sola.

d – infine, non si vede proprio, qui. Non è stupefacente?

5.6 Scatole

Il LaTeX costruisce le pagine manipolando scatole: dapprima, ogni lettera è una piccola scatola, che viene poi incollata ad altre lettere per formare parole; queste sono a loro volta incollate ad altre parole, ma stavolta con una speciale colla elastica, sicché una serie di parole può essere ristretta o stirata in modo da riempire completamente una riga della pagina.

Questa è certamente una versione semplicistica di cosa succede realmente, ma il punto è che il TeX lavora su colla e scatole. Non solo le lettere sono scatole: virtualmente ogni cosa può essere messa in una scatola, anche altre scatole: ogni scatola verrà poi trattata dal LaTeX come se fosse una singola lettera.

Nel capitolo precedente sono già state presentate alcune scatole, sebbene non sia stato detto esplicitamente: l'ambiente `tabular` ed il comando `\includegraphics`, per esempio, producono scatole. Questo significa che si possono facilmente affiancare due tabelle o due immagini: bisogna solo assicurarsi che la larghezza complessiva non superi la larghezza del testo.

Si può anche imballare un paragrafo in una scatola, o con il comando

`\parbox[`*pos*`]{`*largh*`}{`*testo*`}`

o con l'ambiente

`\begin{minipage}[`*pos*`]{`*largh*`}` testo `\end{minipage}`.

Il parametro *pos* è una delle lettere `c`, `t` o `b`, e serve per controllare l'allineamento verticale della scatola, relativamente alla linea di base del testo attorno. *largh* è un argomento di tipo lunghezza che specifica l'ampiezza della scatola. La differenza principale tra `minipage` e `parbox` è che non tutti i comandi si possono usare dentro `parbox`, mentre in una `minipage` si può usare praticamente tutto.

Mentre `\parbox` imballa un intero paragrafo, suddividendo le linee etc, c'è un gruppo di comandi di inscatolamento che lavorano solo su materiale allineato orizzontalmente; uno di questi è già stato introdotto: `\mbox`, che si limita ad imballare una serie di scatole per formarne un'altra, e può essere usato per impedire al LaTeX di separare due parole. Poiché si possono mettere scatole dentro scatole, questi imballatori orizzontali danno la massima flessibilità.

`\makebox[`*largh*`][`*pos*`]{`*testo*`}`

largh imposta la larghezza della scatola, come vista dall'esterno[5]. Oltre ad espressioni di lunghezza, si possono usare i comandi `\width`, `\height`, `\depth` e `\totalheight` nel parametro di larghezza, per impostarla uguale allla larghezza, altezza profondità ed altezza complessiva (rispettivamente) del *testo* da comporre. Il parametro *pos* è una lettera: centrato, `l` allineato a sinistra (*left*), `r` allineato a destra (*right*), o `s` che distribuisce il testo in modo da riempire la scatola (*spread*).

Il comando `\framebox` funziona esattamente come `\makebox`, ma disegna una cornice attorno alla scatola.

Ecco alcune cose che si possono fare con `\makebox` e `\framebox`:

[5]Questo significa che può essere più piccola del materiale nella scatola; si può persino impostare la larghezza a 0pt, in modo che il testo nella scatola venga scritto senza influenzare le scatole vicine.

```
\makebox[\textwidth]{%
    c e n t r a t o}\par
\makebox[\textwidth][s]{%
    d i f f u s o}\par
\framebox[1.1\width]{Credo mi
    abbiano incastrato!} \par
\framebox[0.8\width][r]{Orpo,
    sono troppo largo} \par
\framebox[1cm][l]{fa niente,
    anch'io}
Puoi leggere qui sotto?
```

c e n t r a t o

d i f f u s o

Credo mi abbiano incastrato!

Orpo, sono troppo largo

fa niente, anch'io Puoi leggere qui sotto?

Ora che sappiamo controllare l'orizzontale, il passo logico successivo è passare al verticale[6]. Non è un problema per il LaTeX: il comando

`\raisebox{`*sollevamento*`}[`*profondità*`][`*altezza*`]{`*testo*`}`

permette di definire le proprietà verticali di una scatola; si possono usare `\width`, `\height`, `\depth` e `\totalheight` per i primi tre parametri, in modo da manipolare la dimensione della scatola in rapporto all'argomento *testo*.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa\raisebox{-0.3ex}a%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}r%
\raisebox{-2.2ex}g%
\raisebox{-4.5ex}h}}
grid\'o lui, ma neanche il suo
successore nella riga not\'o che
qualcosa di tremendo gli era
successo.
```

Aaaaaaargh gridò lui, ma neanche il suo successore nella riga notò che qualcosa di tremendo gli era successo.

5.7 Righi e puntelli

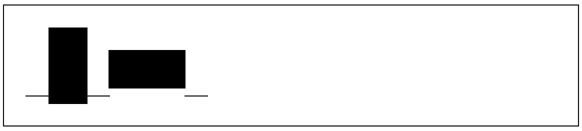
Qualche pagina fa è stato usato il comando

`\rule[`*sollevamento*`]{`*larghezza*`}{`*altezza*`}`,

che normalmente produce un rettangolo nero.

[6] Il controllo totale si può ottenere solo con il controllo sia dell'orizzontale, sia del verticale . . .

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



Questo è molto utile per disegnare linee orizzontali e verticali; ad esempio, la linea nella pagina del titolo è stata creata con un comando `\rule`.

Un caso particolare è un rigo senza larghezza ma con una certa altezza; con terminologia tipografica, questo è chiamato puntello, e viene usato per garantire che un certo elemento in una pagina abbia una certa altezza minima; lo si può ad esempio usare in un ambiente `tabular` per assicurarsi che una riga abbia una certa altezza.

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Trave di
    sostegno\ldots\\
\hline
\rule{0pt}{4ex}Puntello\\
\hline
\end{tabular}
```

Trave di sostegno...
Puntello

Bibliografia

[1] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, 2ª ed., 1994, ISBN 0-201-52983-1.

[2] Donald E. Knuth. *The TeXbook,* Volume A di *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, 2ª ed., 1984, ISBN 0-201-13448-9.

[3] Michel Goossens, Frank Mittelbach e Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.

[4] Ogni installazione di LaTeX dovrebbe offrire una cosiddetta *LaTeX Local Guide*, che spiega le particolarità del sistema su cui gira. Dovrebbe essere contenuta in un file chiamato `local.tex` ma, sfortunatamente, alcuni amministratori negligenti non forniscono questo documento. In tal caso si chieda aiuto al guru locale di LaTeX.

[5] LaTeX3 Project Team. *LaTeX 2ε for authors*. È incluso nella distribuzione di LaTeX 2ε come `usrguide.tex`.

[6] LaTeX3 Project Team. *LaTeX 2ε for Class and Package writers*. È incluso nella distribuzione di LaTeX 2ε come `clsguide.tex`.

[7] LaTeX3 Project Team. *LaTeX 2ε Font selection*. È incluso nella distribuzione di LaTeX 2ε come `fntguide.tex`.

[8] D. P. Carlisle. *Packages in the 'graphics' bundle*. Si trova nel *bundle* 'graphics' come `grfguide.tex`, disponibile presso la stessa fonte della propria distribuzione di LaTeX.

[9] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of LaTeX's verbatim Environments*. Si trova nel *bundle* 'tools' come `verbatim.dtx`, disponibile presso la stessa fonte della propria distribuzione di LaTeX.

[10] Graham Williams. *The TeX Catalogue* è una lista molto accurata di pacchetti collegati a TeX e LaTeX. Disponibile online presso `CTAN:/tex-archive/help/Catalogue/catalogue.html`

[11] Keith Reckdahl. *Using EPS Graphics in LaTeX 2ε Documents*, che spiega tutto e molto più di quanto si possa desiderare conoscere in materia di file EPS e del loro uso nei documenti LaTeX. È disponibile online presso `CTAN:/tex-archive/info/epslatex.ps`