Impara LaTeX!

(...e mettilo da parte)

Marc Baudoin

École Nationale Supérieure de Techniques Avancées

32, boulevard Victor
75739 PARIS cedex 15

Prefazione

A proposito di questo manuale

Questo manuale è inizialmente stato scritto ad uso degli studenti dell'*École nationale supérieure de techniques avancées* (ENSTA) di Parigi, ed è basato sulla versione di LaTeX ivi installata[1].

È possibile che l'installazione di LaTeX sulla vostra rete sia leggermente diversa. In particolare, possono mancare le estensioni babel e graphicx.

Consultate lo staff tecnico incaricato dell'amministrazione della rete per sapere cosa è disponibile sulla rete o per chiedergli di installare quello che manca.

Per approfondire

Questo manuale non vuole essere esaustivo, e descrive solo gli aspetti di LaTeX che mi sembrano essere di maggiore utilità. Per una trattazione più estesa vi consiglio di leggere *LaTeX: a document preparation system*, di Leslie Lamport ([1]). Un altro libro indispensabile è *LaTeX Companion*, di Michel Goossens, Frank Mittelbach e Alexander Samarin ([2]), che descrive numerose estensioni di LaTeX.

Versioni

La prima versione di questo manuale è stata scritta nella primavera 1994 e descriveva LaTeX 2.09. La versione che state leggendo è stata adattata nel 1996 a LaTeX 2_ε.

Dove trovare questo manuale?

Questo manuale è disponibile in vari formati sui seguenti siti tramite FTP anonimo:

- *ftp://ftp.agm-ita.ensta.fr/pub/babafou/*
- *ftp://ftp.pluto.linux.it/pub/pluto/ildp/misc/impara_latex/*

[1]Nell'edizione italiana, l'installazione di riferimento è LaTeX 2_ε quale installato al centro Cefriel di Milano.

Ringraziamo l'*Amicale du Génie Maritime et des Ingénieurs Techniques Avancées (AGM-ITA)* e Michele Dalla Silvestra per ospitare questo manuale sui loro archivi FTP.

Suggerimenti, commenti e complimenti

Mi farebbe piacere sapere cosa ne pensate del manuale (nel bene e nel male). Per farlo, potete inviare un messaggio[2] all'indirizzo seguente:

`<babafou@agm-ita.ensta.fr>`

[2] In inglese o francese...

Nota all'edizione italiana

Questa prima edizione italiana di *Apprends LaTeX !* è ancora in fase di redazione. Oltre alla traduzione, sono stati effettuati i necessari adattamenti (regole tipografiche, estensioni per l'italiano e così via) alla lingua italiana ed alle configurazioni di LaTeX più in uso in Italia.

Per eventuali commenti, suggerimenti, critiche, correzioni, non fatevi problemi a contattarmi all'indirizzo seguente:

`<cannarsi@cefriel.it>`

Capitolo 1

Introduzione

Questo manuale ha lo scopo di insegnarvi LaTeX[1] con un po' di humour. Dopo averlo letto dovreste essere in grado di utilizzare LaTeX per tutti i vostri documenti.

1.1 Cos'è questo LaTeX?

LaTeX è un formattatore di testi molto potente, che produce documenti di eccellente qualità. È utilizzato da moltissimi studenti e ricercatori in tutto il mondo.

Per avere un'idea di ciò che è capace di fare, sfogliate questo manuale: è stato interamente realizzato con LaTeX.

Il vantaggio di LaTeX rispetto ad altri word processor è che dispone di un vero e proprio linguaggio tramite il quale implementare nuove funzioni.

LaTeX è un'estensione di un formattatore di testi chiamato TeX (il nome TeX viene dal greco $\tau\acute{\varepsilon}\chi\nu\eta$[2], che vuol dire "arte").

1.2 Convenzioni tipografiche

Per facilitare la lettura di questo documento sono state utilizzate le seguenti convenzioni tipografiche:

- lo stile `macchina da scrivere` è utilizzato per gli esempi di codice LaTeX, per i nomi dei comandi LaTeX e per i comandi UNIX introdotti da tastiera;
- il *corsivo* è utilizzato nella definizione di nuovi termini, per le opzioni e gli argomenti dei comandi LaTeX così come per i nomi di file e di comandi UNIX;
- lo stile sans serifs è utilizzato per le classi e le loro opzioni, le estensioni e le loro opzioni, nonché per gli ambienti.

[1] Nulla a che vedere col caucciù. Si pronuncia "làtec".
[2] Da cui viene "tecnica": ecco spiegata la pronuncia di LaTeX.

Capitolo 2

Perché imparare LaTeX?

Avete deciso di imparare ad utilizzare LaTeX: bravi! Ma certamente vi chiederete quali vantaggi offre rispetto ad un word processor come Word. In effetti LaTeX ha la cattiva fama di un programma difficile da apprendere, mentre Word è molto più semplice e si usa in modo intuitivo (chi ha mai letto i manuali di Word?). Ecco a mio avviso i principali punti a favore di LaTeX:

- le funzioni complesse, come la realizzazione di un indice o dell'indice analitico, sono difficili da realizzare con Word mentre basta un solo comando in LaTeX;
- LaTeX è particolarmente adatto ai testi che contengono formule matematiche;
- LaTeX è estendibile (gli si possono aggiungere nuove funzioni).

Chiaramente LaTeX ha anche dei difetti, ma li vedremo dopo, quando ne saprete un po' di più.

2.1 La filosofia di LaTeX

LaTeX non è un word processor propriamente detto, come Word; è un *formattatore di testi*. Un documento LaTeX è un file di testo scritto con un qualsiasi editor (*Emacs*, *vi*...) che contiene, oltre al testo, dei comandi di formattazione. Quindi, primo punto importante, non si vede il documento così come verrà stampato mentre lo si scrive. Si dice che LaTeX non è *WYSIWYG* (*What You See Is What You Get*), contrariamente a Word. Per visualizzare il documento bisogna compilarlo con LaTeX, poi utilizzare un programma di visualizzazione. È questo il grosso inconveniente di LaTeX.

D'altra parte, siccome tratta il testo nel suo insieme (e non man mano che lo si redige), può avere una visione d'insieme e fare scelte di impaginazione migliori.

Di più, LaTeX "vede" il testo come lo pensiamo mentre Word vede il testo come lo battiamo. Tutto il vantaggio di LaTeX sta in questa "intelligenza".

2.2 Come funziona LATEX?

Quando compila un documento, LATEX genera un certo numero di file:

- un file `.dvi` (*device-independent*) che è il risultato della compilazione: questo file servirà alla stampa del documento (dopo essere stato convertito in PostScript);
- un file `.aux` contenente le cross-reference;
- un file `.log` contenente una copia dei messaggi che appaiono durante la compilazione.

A seconda della complessità del vostro documento, LATEX può creare anche i file:

- `.toc`, che contiene l'indice (*table of contents*);
- `.lot`, che contiene la lista delle tabelle (*list of tables*) ;
- `.lof`, che contiene la lista delle figure (*list of figures*) ;
- `.idx`, che contiene l'indice analitico non formattato.

Capitolo 3

Il vostro primo documento

Cominciamo dall'inizio. In questo capitolo realizzerete il vostro primo documento con LaTeX, imparando a:

- compilare un documento;
- trovare e correggere gli errori che può contenere;
- visualizzarlo;
- stamparlo.

3.1 Prima di cominciare

Create una directory `tex` nella quale metterete i vostri testi. Fatto ciò, potrete realizzare il vostro primo documento con LaTeX.

3.2 Cominciamo!

3.2.1 L'editing

Create il vostro documento con il vostro editor preferito (*Emacs*, *vi*...) e scrivete:

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

\usepackage[latin1]{inputenc}
\usepackage[italian]{babel}

\begin{document}

Questo è il mio primo documento con \LaTeX.

\end{document}
```

Salvate il testo con il nome `primo.tex` (per convenzione, i documenti LaTeX hanno l'estensione `.tex`).

3.2.2 La compilazione

Per compilare il vostro documento utilizzate il comando

```
latex primo.tex
```

LaTeX mostra man mano quello che sta facendo:

```
This is TeX, Version 3.14159 (C version 6.1)
(primo.tex
LaTeX2e <1995/12/01> patch level 2
Hyphenation patterns for english, english, italian, loaded.
(/opt/TeX/texmf/tex/latex/base/article.cls
Document Class: article 1995/12/20 v1.3q Standard LaTeX document class
(/opt/TeX/texmf/tex/latex/base/size11.clo))
(/opt/TeX/texmf/tex/latex/base/inputenc.sty beta test version
(/opt/TeX/texmf/tex/latex/base/latin1.def))
(/opt/TeX/texmf/tex/generic/babel/babel.sty
(/opt/TeX/texmf/tex/generic/babel/italian.ldf
(/opt/TeX/texmf/tex/generic/babel/babel.def)))
No file primo.aux.
[1] (primo.aux) )
Output written on primo.dvi (1 page, 344 bytes).
Transcript written on primo.log.
```

3.2.3 Rilevamento e correzione degli errori

Se vi siete sbagliati da qualche parte, appare un messaggio di questo tipo:

```
! Undefined control sequence.
l.9  Questo è il mio primo documento con \latex
                                               .

?
```

Quando trova un errore, LaTeX indica:

- la natura dell'errore (`Undefined control sequence`) ;
- la linea in cui si trova (`l.9`) ;
- così come il punto preciso dell'errore nella linea (per indicarlo, LaTeX torna a capo): qui il comando sconosciuto è `\latex`.

In questo caso, battete `x` oppure `^D` per interrompere la compilazione, correggete l'errore nel testo, salvate il file e ricompilate.

3.2.4 La visualizzazione

Per visualizzare il documento, battete

```
xdvi primo.dvi
```

Si aprirà una finestra che mostra il documento così come verrà stampato.

Con i bottoni situati nella parte destra della finestra potete:

- spostarvi all'interno del codumento (cosa ugualmente possibile da tastiera: la barra spaziatrice permette di passare alla pagina successiva ed il tasto $\boxed{\text{BackSpace}}$ alla precedente);
- modificare l'ingrandimento (*shrink*).

È altresì possibile effettuare uno zoom cliccando nella finestra con i bottoni del mouse:

bottone di sinistra	→	zoom piccolo
bottone di centro	→	zoom medio
bottone di destra	→	zoom grande

3.2.5 La stampa

Per stampare il documento bisogna innanzitutto convertirlo in PostScript (è il linguaggio di descrizione testi utilizzato dalle stampanti laser) tramite il comando

```
dvips primo.dvi -o
```

Non resta altro che stampare il file così creato, `primo.ps` con il comando *lpr* (o *lp*, a seconda del sistema operativo).

Capitolo 4

I concetti base di LaTeX

Per poter utilizzare appieno LaTeX e sfruttare tutte le sue potenzialità, bisogna capire alcune idee base. Questo capitolo vi inizierà più approfonditamente alla filosofia ed allo spirito di LaTeX e vi spiegherà l'esempio del capitolo precedente.

4.1 Il documento LaTeX

Un documento LaTeX è un insieme di testo e comandi di formattazione.
Riprendiamo l'esempio precedente:

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

\usepackage[latin1]{inputenc}
\usepackage[italian]{babel}

\begin{document}

Questo è il mio primo documento con \LaTeX.

\end{document}
```

Come avete già avuto modo di vedere, tutto questo serve solo a stampare il testo: "Questo è il mio primo documento con LaTeX." Ma attenzione, il resto è tutt'altro che inutile! Sono i famosi comandi di formattazione. Esaminiamo il primo:

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

Questo comando è indispensabile e deve comparire nella prima linea di tutti i documenti LaTeX. Definisce la *classe* del documento (qui article) e le *opzioni* da applicare (qui a4paper e 11pt).

4.1.1 Sintassi di un comando LaTeX

La sintassi generale di un comando LaTeX è la seguente:

`\comando[`*opzione*`]{`*parametro*`}`

- un comando inizia con un backslash `\`;
- il backslash è seguito dal nome del comando;
- seguono poi i parametri opzionali fra parentesi quadre;
- ed infine i parametri obbligatori tra parentesi graffe.

Ecco qualche esempio di comando:

- comando senza parametri:
  `\appendix`
- comando con parametri:
  `\chapter{nome}`
- comando con parametri opzionali:
  `\item[nome]`
- comando con parametri opzionali ed obbligatori:
  `\cite[pag. 150]{lamport-latex}`

4.1.2 Il comando `\documentclass`

Ritorniamo al comando `\documentclass` ed ai suoi parametri.

La classe di un documento

La classe, che è il parametro obbligatorio del comando `\documentclass`, definisce il tipo del documento. Esistono cinque classi:

article è la classe utilizzata per i documenti più corti (articoli per riviste scientifiche...);

report è utilizzata per i documenti più lunghi che devono essere divisi in capitoli (mentre la classe articole non gestisce i capitoli);

book è utilizzata per i libri (ci sono poche differenze rispetto alla classe report, la più importante è che questa classe, per default, produce documenti fronte-retro);

letter permette di scrivere delle lettere;

slides serve invece a fare dei lucidi.

Le opzioni di classe

Le opzioni di classe, che sono i parametri opzionali del comando `\documentclass`, servono a modificare l'impaginazione del documento.

Spieghiamo un attimo quelle utilizzate qui:

a4paper permette di impostare i margini del documento per una stampa su carta in formato A4 (quest'opzione è indispensabile, altrimenti vi ritroverete con dei margini per carta americana);

11pt porta la grandezza dei caratteri a undici punti[1] (la grandezza normale, senza opzioni, è di dieci punti; esiste anche un'opzione 12pt).

4.1.3 Le estensioni

Le estensioni (*package* in inglese) servono a modificare l'impaginazione o a definire nuovi comandi, e quindi ad aggiungere nuove funzionalità a LaTeX. Vengono caricate tramite il comando `\usepackage` nel preambolo del documento (v. § 4.1.4).

Alcune estensioni sono distribuite con LaTeX, altre devono essere installate dall'amministratore del pacchetto. Tra le estensioni standard, troviamo:

fontenc che, utilizzata con l'opzione T1, dice a LaTeX di utilizzare la codifica di caratteri T1 (è il nuovo standard LaTeX per la codifica dei caratteri ma non è impostato di default per motivi di compatibilità con le versioni precedenti, da cui la necessità di utilizzare questa estensione quando possibile scrivendo nel preambolo[2]:

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

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[italian]{babel}

\begin{document}
```

inputenc che permette, grazie all'opzione latin1, di utilizzare dei caratteri ISO-8859-1 nel documento. Con questa estensione potete battere e vedere direttamente i caratteri accentati nel documento: vi raccomando di utilizzarla sistematicamente (per l'introduzione di caratteri accentati v. oltre).

L'estensione babel, anche se non è standard, è molto utile. Combinata all'opzione italian adatta LaTeX alle convenzioni tipografiche italiane e ridefinisce alcuni titoli in

[1] Il punto è l'unità di misura in tipografia. Un punto vale 1/72esimo di pollice (1 pollice = 2,54 cm).

[2] La codifica T1 potrebbe non essere installata sulla vostra rete (per esempio al Cefriel) e dare problemi in fase di compilazione, con dei messaggi d'errore che assomigliano a: `! Font T1/cmr/m/it/10.95=dcti1095 at 10.95pt not loadable: Metric (TFM) file not found.` in questo caso, non includere quest'opzione.

italiano, come "Indice" o "Capitolo" (molto meglio di "Table of contents" e "Chapter", se il vostro documento è in italiano).

Esistono moltissime altre estensioni, alcune delle quali verranno trattate in seguito.

4.1.4 Il preambolo

Il *preambolo* è la parte del sorgente compresa tra i comandi `\documentclass` e `\begin{document}`. Nel nostro esempio, contiene i comandi che caricano le estensioni.

4.1.5 Il documento

Il documento vero e proprio è contenuto tra `\begin{document}` e `\end{document}`. Tutto ciò che è scritto dopo `\end{document}` viene ignorato.

Qui incontriamo una struttura molto utilizzata in LaTeX: la struttura d'*ambiente* (v. cap. 6 per una descrizione completa degli ambienti di LaTeX).

4.2 Struttura del testo

4.2.1 I paragrafi

Un documento LaTeX è formato di *paragrafi* separati l'uno dall'altro tramite una o più linee bianche:

```
Ecco un paragrafo molto breve.

Qui comincia un altro paragrafo. Come potete vedere,
i ritorni a capo non hanno alcuna importanza nella
formattazione del paragrafo: contano solo le linee bianche.

Quest'ultimo paragrafo chiude l'esempio.
```

Ed ecco il risultato:

> Ecco un paragrafo molto breve.
>
> Qui comincia un altro paragrafo. Come potete vedere, i ritorni a capo non hanno alcuna importanza nella formattazione del paragrafo: contano solo le linee bianche.
>
> Quest'ultimo paragrafo chiude l'esempio.

4.2.2 Le parole

Un paragrafo è composto di *parole* separate da segni di punteggiatura o spazi (un ritorno a capo è considerato come uno spazio). Potete inserire quanti spazi volete tra due parole (anche se ne basta uno). Per esempio i due testi che seguono danno lo stesso risultato:

```
Ecco un esempio di testo, con qualche spazio.
```

```
Ecco        un           esempio
   di     testo,          con
       qualche spazio.
```

Infine, un piccolo consiglio. Dal momento che il ritorno a capo è equivalente ad uno spazio, non scrivete linee di testo di più di 80 caratteri: andate a capo. Infatti è più facile spostarsi con i tasti cursore in un paragrafo composto di più linee piuttosto che in un paragrafo formato da una sola. In più, alcuni editor vanno a capo da soli una volta raggiunto il margine destro (questa funzionalità viene spesso chiamata *wrap long lines*. Se utilizzate Emacs, potete mettere le seguenti linee nel vostro file di configurazione `.emacs` per attivare automaticamente il ritorno a capo automatico quando siete in modo LATEX:

```
;; Per andare a capo automaticamente in modo latex
(add-hook 'latex-mode-hook 'auto-fill-mode)
```

4.2.3 Lo spazio insecabile: ~

In alcuni casi bisogna impedire che LATEX separi due parole alla fine della linea, come accade qui:

> Qual è il nome dell'attore che in "Dalla Russia con amore" interpreta l'agente
> 007?

Lo "007" all'inizio della linea è estremamente sgradevole.

Vorremmo poter appiccicare lo "007" alla parola che lo precede: bisogna usare il carattere~. Questo carattere rappresenta uno *spazio insecabile*, che viene considerato come uno spazio (e quindi apparirà come tale in stampa), ma LATEX non separerà mai in fine linea le due parole:

```
Qual è il nome dell'attore che in ``Dalla Russia con amore''
interpreta l'agente~007?
```

> Qual è il nome dell'attore che in "Dalla Russia con amore" interpreta l'a-
> gente 007?

Non utilizzate sistematicamente lo spazio insecabile, perché rende il testo meno leggibile. Quando avrete terminato la redazione del vostro documento potrete eventualmente inserire spazi insecabili laddove è necessario.

4.2.4 I commenti

LATEX permette di includere dei commenti nel testo, che non verranno stampati. Possono servire come punti di riferimento per l'autore, ricordargli che non ha finito di scrivere un paragrafo ecc.

Un commento comincia con `%` e termina alla fine della linea:

```
Questo è testo normale.       % questo è un commento
Questo è il resto del testo.
```

4.3 Caratteri speciali utilizzati da LaTeX

Avrete notato che alcuni caratteri hanno un significato speciale per LaTeX. Ad esempio, `\` indica un comando, `~` uno spazio insecabile e `%` un commento.

Esistono dieci caratteri speciali:

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

che non possono essere stampati così come sono.

Per i primi sette esiste un comando speciale, che consiste in un `\` (perché è un comando) seguito dal carattere in questione. Quindi,

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

stampano

$ & % # _ { }

Per gli altri tre, LaTeX non fornisce comandi che permettano di stamparli, ma si può aggirare l'ostacolo utilizzando le combinazioni `\~{}`, `\^{}` e `$\backslash$` che permettono di stampare rispettivamente ˜ e ˆ.

4.4 I problemi di cesura: `Overfull \hbox`

A volte capita che LaTeX non sappia dove dividere una parola alla fine di una linea; e piuttosto che dividerla a caso, preferisce non far nulla. In questo caso la parola in questione oltrepassa il margine destro, e LaTeX genera il seguente messaggio:

```
Overfull \hbox (8.45247pt too wide) in paragraph at lines 35--39
```

Il messaggio di warning `Overfull \hbox` significa che qualcosa oltrepassa il margine destro. LaTeX segnala di quanto spazio il testo oltrepassa il margine (`8.45247pt`) e in quali linee si trova il paragrafo che dà problemi (`lines 35--39`).

Non preoccupatevi: capita molto raramente (in questo manuale, ho avuto solo 8 di questi messaggi), e quasi sempre con delle parole che contengono una lettera accentata.

Quando siete in questa situazione, è possibile precisare dove dividere la parola (*cesura*). Per farlo, trovate il punto preciso con *xdvi* e inserite nel testo il comando `\-` esattamente nel punto in cui volete dividere la parola.

Il comando `\-` ha effetto sulla formattazione del resto del paragrafo, perché una parte della parola si ritroverà sulla linea seguente e sposterà il resto del testo. Può darsi allora che compaiano o scompaiano altri avvertimenti di cesura. Per questo motivo vi consiglio di effettuare le cesure una per una, verificando di volta in volta il risultato (lo si può fare visualmente con *xdvi*, ma è sufficiente che il messaggio `Overfull \hbox` scompaia per essere sicuri di aver effettuato la cesura come si deve) e solamente quando avete finito di scrivere il paragrafo.

Capitolo 5

I comandi di LaTeX

I comandi sono utilizzati di frequente con LaTeX: per stampare alcuni caratteri che non sono sulla tastiera, per modificare la tipografia, per strutturare il documento ecc.

5.1 Sintassi di un comando

La sintassi generale di un comando LaTeX è la seguente:

`\comando[`*opzione*`]{`*parametro*`}`

- un comando inizia con un backslash `\`;
- il backslash è seguito dal nome del comando;
- seguono poi i parametri opzionali fra parentesi quadre;
- ed infine i parametri obbligatori tra parentesi graffe.

Alcuni comandi hanno ciò che viene detto *variante asterisco* , cioè una variante del comando. Per ottenere la variante asterisco di un comando è sufficiente aggiungere un asterisco tra il nome del comando e i suoi parametri:

`\comando*[`*opzione*`]{`*parametro*`}`

5.2 Caratteri speciali

Certi caratteri, ed alcuni simboli, non si trovano sulla tastiera. Per coloro che utilizzano tastiere QWERTY, in particolare, è il caso delle lettere accentate, indispensabili in italiano. Fortunatamente LaTeX mette a vostra disposizione una serie di comandi che colmano questa lacuna.

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

Tabella 5.1: Simboli di uso comune

œ	`\oe`	å	`\aa`	ł	`\l`	¿	`?‘`
Œ	`\OE`	Å	`\AA`	Ł	`\L`	¡	`!‘`
æ	`\ae`	ø	`\o`	ß	`\ss`		
Æ	`\AE`	Ø	`\O`				

Tabella 5.2: Simboli per l'italiano e le altre lingue

5.2.1 Simboli

LATEX definisce dei comandi che permettono di stampare diversi simboli (v. tab. 5.1) di uso comune ed altri utilizzati in italiano e in altre lingue (v. tab. 5.2).

Segnaliamo inoltre la presenza del comando `--` che stampa un trattino utilizzato per gli intervalli di numeri (come in 93–94) e del comando `---` che stampa una lineetta utilizzata per la punteggiatura — come questa.

LATEX definisce altresì tre altri comandi che permettono di stampare i logo TEX (`\TeX`), LATEX (`\LaTeX`) e LATEX 2$_\varepsilon$ (`\LaTeXe`).

Vediamo ora ciò che risulta dal testo seguente:

```
Prova dei comandi \TeX e \LaTeX in un testo.
```

Ecco il risultato:

Prova dei comandi TEXe LATEXin un testo.

Non è precisamente quello che ci aspettavamo. In effetti, LATEX ignora tutti gli spazi (uno o più) che seguno un comando senza parametri (ed è il caso dei comandi `\TeX`, `\LaTeX` e `\LaTeXe`). Possiamo cavarcela in tre modi:

- utilizzare il comando `\␣` (backslash spazio) che stampa uno spazio:

  ```
  Prova dei comandi \TeX\ e \LaTeX\ in un testo.
  ```

- mettere un parametro vuoto:

  ```
  Prova dei comandi \TeX{} e \LaTeX{} in un testo.
  ```

- mettere il comando tra parentesi graffe (vengono chiamate *graffe di raggruppamento*: v. pag. 17) :

  ```
  Prova dei comandi {\TeX} e {\LaTeX} in un testo.
  ```

Tutti questi metodi sono equivalenti (danno esattamente lo stesso risultato) ma il primo è più elegante (mettere un parametro vuoto ad un comando che non ne ha non è molto logico) e il sorgente risulta più leggibile. Quanto all'ultimo, viene utilizzato soprattutto per i simboli della tab. 5.2. Per esempio "perché" si scrive `perch\'{e}`.

5.2.2 Le lettere accentate

LATEX definisce dei comandi che permettono di stampare ogni tipo di lettera accentata (v. tab. 5.3). Beninteso, potete anche batterle nel vostro editor (a condizione di utilizzare l'estensione `inputenc` con l'opzione `latin1`).

è	`\`{e}`	ñ	`\~{n}`	ě	`\v{e}`	ç	`\c{c}`
é	`\'{e}`	ē	`\={e}`	e̋	`\H{e}`	ẹ	`\d{e}`
ê	`\^{e}`	ė	`\.{e}`	e͡e	`\t{ee}`	e̱	`\b{e}`
ë	`\"{e}`	ĕ	`\u{e}`				

Tabella 5.3: Accenti

Tutto questo però non è molto pratico. In LATEX si possono allora utilizzare delle abbreviazioni per alcuni di questi comandi, presentate nella tab. 5.4.

è	`\`e`	ñ	`\~ n`
é	`\'e`	ē	`\=e`
ê	`\^ e`	ė	`\.e`
ë	`\"e`		

Tabella 5.4: Abbreviazioni dei comandi d'accentazione

Ecco un esempio di testo in italiano che utilizza caratteri accentati:

```
Non se ne pu\`o pi\`u, in realt\`a, cos\`{\i} facendo
il numero di caratteri da battere \`e triplo! Perch\'e
non usare i caratteri ISO-8859-1?
```

ATTENZIONE: per mettere accenti sulle lettere "i" e "j" bisogna utilizzare i comandi `\i` e anziché le lettere: in questo modo sarà possibile stamparle senza puntino. Date un'occhiata qui:

```
\`i \`e leggermente diverso da \`{\i}.
```

ì è leggermente diverso da ì.

Se invece utilizzate i caratteri ASCII estesi (ISO-8859-1) come spiegato nel paragrafo successivo, non c'è problema.

Per chi ha una tastiera QWERTY ed utilizza Emacs come editor, una soluzione ancora più pratica è utilizzare l'*iso-accents-mode*. In questo modo ogni lettera accentata viene introdotta in due fasi: prima l'accento e poi la lettera. Per esempio, per "à" bisogna battere prima `` ` `` e poi A. Una volta attivato questo modo, sarà ancora possibile, ovviamente, inserire accenti da soli (apostrofo, circonflesso): basterà battere uno spazio dopo l'accento in questione affinché non venga considerato come un carattere speciale. In *iso-accents-mode* quindi, "^" si scrive battendo ^ seguito dalla barra spaziatrice.

Inserite queste linee nel vostro file `.emacs` per attivare automaticamente questo modo quando selezionate il modo LATEX:

```
;; Per visualizzare i caratteri ISO-8859-1
(standard-display-european t)

;; Per introdurre i caratteri ISO-8859-1 da tastiera
(add-hook 'latex-mode-hook 'iso-accents-mode)
```

Un altro modo è riprogrammare i tasti funzione tramite il comando UNIX *xmodmap*.

5.3 Dichiarazioni e comandi di cambiamento di stile

LATEX permette ovviamente di cambiare lo stile dei caratteri. In tipografia, gli stili si distinguono:

- per la forma (v. tab. 5.5) ;

`{\upshape Upright shape}`	Upright shape
`{\itshape Italic shape}`	*Italic shape*
`{\slshape Slanted shape}`	*Slanted shape*
`{\scshape Small capitals}`	SMALL CAPITALS

Tabella 5.5: Dichiarazioni di cambiamento di forma

- il peso (v. tab. 5.6) ;

`{\mdseries normale}`	normale
`{\bfseries grassetto}`	**grassetto**

Tabella 5.6: Dichiarazioni di cambiamento di peso

- la famiglia (v. tab. 5.7).

Ognuna di queste caratteristiche può essere cambiata indipendentemente dalle altre.

`{\rmfamily famiglia roman}`	famiglia roman
`{\sffamily famiglia sans serifs}`	famiglia sans serifs
`{\ttfamily famiglia typewriter}`	`famiglia typewriter`

Tabella 5.7: Dichiarazioni di cambiamento di famiglia

I comandi qui indicati sono diversi da quelli che abbiamo visto finora. Non hanno argomenti ed agiscono su tutto il resto dell testo. Le si chiama *dichiarazioni*.

Al fine di limitare la visibilità di una dichiarazione bisogna utilizzare delle *graffe di raggruppamento*:

```
Metto la parola che segue in {\itshape corsivo}.
Il resto del testo è normale.
```

Metto la parola che segue in *corsivo*. Il resto del testo è normale.

Si possono anche combinare più dichiarazioni:

```
Io mi {\itshape chiamo {\bfseries Gigi}}.
```

Io mi *chiamo **Gigi***.

Già che ci siamo, noterete che gli spazi che seguono una dichiarazione vengono ignorati (come gli spazi che seguivano i comandi `\TeX` e `\LaTeX`).

Esiste un comando corrispondente a ciascuna di queste dichiarazioni (v. tab. 5.8).

Dichiarazione	Comando
`\upshape`	`\textup`
`\itshape`	`\textit`
`\slshape`	`\textsl`
`\scshape`	`\textsc`
`\mdseries`	`\textmd`
`\bfseries`	`\textbf`
`\rmfamily`	`\textrm`
`\sffamily`	`\textsf`
`\ttfamily`	`\texttt`

Tabella 5.8: Comandi di cambiamento di stile

Questi comandi modificano unicamente lo stile del loro parametro, mentre il testo che segue non viene modificato:

```
Metto la parola che segue in \textit{corsivo}.
Il resto del testo è normale.
```

Metto la parola che segue in *corsivo*. Il resto del testo è normale.

Si raccomanda, per quanto possibile, di adoperare i comandi di cambiamento di stile piuttosto che le dichiarazioni equivalenti.

Così come le dichiarazioni possiamo anche combinare i comandi:

```
Io mi \textit{chiamo \textbf{Gigi}}.
```

Io mi *chiamo **Gigi***.

Segnaliamo anche il comando `\emph`, che serve ad evidenziare il suo parametro:

```
Evidenziamo \emph{queste parole}.
```

Evidenziamo *queste parole*.

E qui incominciamo a vedere uno degli aspetti più importanti della filosofia di LaTeX: la distinzione tra la struttura del documento e la sua "resa" visiva. Come potete vedere, il comando `\emph` evidenzia il suo parametro stampandolo in corsivo. Se questo manuale fosse stato stampato su una stampante che non dispone dello stile corsivo, il parametro di `\emph` sarebbe stato sottolineato. Il comando `\emph`, dunque, non descrive il modo in cui quelle parole devono essere *stampate*, ma il loro valore, il modo in cui le si sono *pensate* nell'ambito del testo.

Immaginate di dover scrivere un libro di botanica e di volere che, nel vostro libro, tutti i nomi latini delle piante appaiano in corsivo. Nulla di più semplice: basta metterle come parametro del comando `\textit`. Poi, quando il vostro libro è finito, il vostro editore vi chiede di mettere i nomi latini in grassetto piuttosto che in corsivo, perché queste sono le sue scelte tipografiche. Bisogna allora sostituire tutti i `\textit` in `\textbf`, e facendo attenzione, perché probabilmente avrete utilizzato il corsivo anche per qualcos'altro. Insomma, non è il massimo della praticità...

LaTeX, conformemente alla sua filosofia, offre una soluzione elegante per questo problema. Infatti è possibile definire dei nuovi comandi (per dettagli, fate riferimento al § 9.1.1). Vi basta dunque definire un comando `\latino` che stamperà il suo parametro, cioè un nome latino, nello stile che preferite. Se i nomi latini devono poi essere in grassetto, basterà allora cambiare la definizione del comando `\latino`, e il gioco è fatto.

Ecco perché non si usano quasi mai dei comandi di cambiamento di stile in un documento LaTeX. Si preferisce definire un insieme di comandi che rispecchiano la logica del testo ed utilizzare questi ultimi. Così per scrivere questo manuale, ho definito vari comandi:

- `\comando` permette di scrivere i comandi in stile macchina da scrivere;
- `\classe` permet di scrivere i nomi delle classi in stile sans serifs;
- e così via.

Vi consiglio fortemente di fare lo stesso.

5.4 Dichiarazioni di cambiamento di corpo

LATEX dispone di dichiarazioni (da utilizzare dunque con delle graffe di raggruppamento) che permettono di modificare il corpo (grandezza dei caratteri) del testo (v. tab. 5.9).

corpo	`\tiny`	corpo	`\large`
corpo	`\scriptsize`	corpo	`\Large`
corpo	`\footnotesize`	corpo	`\LARGE`
corpo	`\small`	corpo	`\huge`
corpo	`\normalsize`	corpo	`\Huge`

Tabella 5.9: Dichiarazioni dei cambiamenti di corpo

5.5 Comandi di interruzione linea, interruzione pagina e di spaziatura

Alcuni comandi più o meno utili possono obbligare LATEX a effettuare particolari scelte di impaginazione del documento.

Il comando `\newline` forza un'interruzione di linea. Il comando `\\` è un'abbreviazione di `\newline`, e lo incontrete spesso. Non ne abusate. In particolare, questo comando *non* serve a passare ad un nuovo paragrafo (anche se molta gente, purtroppo ed erroneamente, lo usa per questo, i paragrafi vanno separati con *una linea bianca*): infatti se terminate un paragrafo con `\\`, la prima linea del paragrafo successivo *non* avrà un rientro.

Il comando `\newpage` permette di inserire un'interruzione di pagina. Il comando `\clearpage` è equivalente, ma in più obbliga LATEX a stampare tutte le tabelle e le figure "volanti" (v. § 6.9).

I comandi `\hspace`{*larghezza*} e `\vspace`{*larghezza*} inseriscono uno spazio (orizzontale per `\hspace` e verticale per `\vspace`) di larghezza {*larghezza*}.

Le larghezze possono essere espresse in una delle seguenti unità di misura:

cm centimetro;

mm millimetro;

in pollice (inch), 1 in = 2,54 cm;

pt punto (1/72esimo di pollice), un'unità tipografica;

pc pica (1 pica = 12 punti), un'altra unità tipografica;

em 1 em è pari alla larghezza della lettera "M" nel font corrente, ed è un'unità speciale di LATEX;

ex 1 ex è pari alla larghezza della lettera “x” nel font corrente, ed è un'unità speciale di LaTeX.

Il comando `\hspace` non è quasi mai utilizzato, al contrario del comando `\vspace` che serve molto spesso a lasciare uno spazio in un testo o in una figura, ad esempio per incollarci (con della colla *vera*) un disegno o uno schema. Lo spazio di 3 centimetri della fig. 5.1 è stato ottenuto col comando:

```
\vspace{3cm}
```

Figura 5.1: Figura vuota

Infine, i comandi `\bigskip`, `\medskip` e `\smallskip` inseriscono uno spazio verticale più o meno grande. Sono molto comode perché la grandezza dello spazio che generano è funzione del font utilizzato.

5.6 Comandi di sezionamento del documento

Strutturare un documento (organizzarlo in parti, capitoli, sezioni, sottosezioni...) è una delle cose più facili da fare con LaTeX. Bisogna usare i comandi della tab. 5.10.

`\part`{*nome*}	`\subsection`{*nome*}
`\chapter`{*nome*}	`\subsubsection`{*nome*}
`\section`{*nome*}	

Tabella 5.10: Comandi di sezionamento

Il comando `\chapter` non è disponibile nella classe article: ecco perché si utilizzano report o book quando si ha bisogno di fare dei capitoli e la classe article quando non se ne ha bisogno.

Questi comandi sono semplici e potenti. Per esempio il titolo di questa sezione è stato ottenuto così:

```
\section{Comandi di sezionamento del documento}
```

Di più, è LaTeX che gestisce automaticamente la numerazione: così è possibile inserire dei capitoli, delle sezioni o modificare il loro ordine senza preoccuparsi del cambiamento dei numeri.

Analogamente, il comando `\tableofcontents` stampa l'indice. Per aggiornare l'indice c'è bisogno di due compilazioni successive: la prima scrive il contenuto dell'indice nel file `.toc`[1] e la seconda lo include nel documento.

Esistono ugualmente i comandi `\listoftables`, che stampa la lista delle tabelle e `\listoffigures`, che stampa la lista delle figure (v. § 6.9). Anche per aggiornare queste liste bisogna compilare due volte (entrano in gioco i file `.lot` e `.lof`).

Varianti asterisco

I comandi di sezionamento hanno una variante asterisco che non numera i titoli, non modifica la numerazione delle altre sezioni e non genera una linea nellíndice. Sono più che altro utilizzate al posto dei comandi normali nei documenti molto brevi, nei quali la numerazione non è necessaria.

Il titolo qui sopra è stato ottenuto con il comando:

```
\subsection*{Varianti asterisco}
```

Ciononostante si può far apparire il titolo nell'indice con il comando:

`\addcontentsline{toc}`{*livello*}{*titolo*}

che farà apparire *titolo* nell'indice a livello *livello* (che può essere `part`, `chapter`, `section`, `subsection` o `subsubsection`). Per esempio, per includere il titolo qui sopra nell'indice avrei potuto scrivere:

```
\addcontentsline{toc}{subsection}{Varianti asterisco}
```

Appendici

Infine, il comando `\appendix` permette di dire a LATEX che i capitoli (per le classi report et book) o le sezioni (per la classe article) che seguono devono essere considerate appendici e contrassegnate con A, B, ecc.

5.7 Il frontespizio

LATEX permette di stampare un titolo a partire da alcune informazioni contenute nel preambolo:

`\title`{***titolo***} specifica il titolo del documento. Si può utilizzare `\\` per andare a capo.

`\author`{***nome***} specifica il nome dell'autore. Se ce n'è più di uno, i nomi devono essere separati da `\and`. Si può usare `\\` per andare a capo.

`\date`{***date***} specifica la data del documento.

[1] toc per *table of contents.*

I comandi `\title` e `\author` sono obbligatori. Se non si utilizza il comando `\date` viene automaticamente inserita la data in cui avviene la compilazione.

Se non si vuole nessuna data basta scrivere:

```
\date{}
```

Per stampare il titolo bisogna richiamare il comando `\maketitle` subito dopo `\begin{document}`.

Ecco un esempio di documento con titolo:

```
\documentclass[a4paper,11pt]{report}

\usepackage[latin1]{inputenc}
\usepackage[italian]{babel}

\title{Questo è il titolo \\ e il suo seguito}
\author{Harley \and Davidson}
\date{26 febbraio 1994}

\begin{document}

\maketitle

...
```

Si possono mettere note al titolo, all'autore od alla data grazie al comando `\thanks`, che inserisce una nota a piè di pagina:

```
\title{Titolo con nota\thanks{ecco la nota}}
```

Per le classi report e book, il titolo è stampato da solo al centro del frontespizio, mentre per la classe article, viene stampato all'inizio della prima pagina. In questo caso, per ottenere un frontespizio su una pagina a sé, bisogna includere l'opzione di classe titlepage.

Questo tipo di titolo, bisogna ammetterlo, è abbastanza spartano, ma è possibile fare da soli l'impaginazione con l'ambiente `titlepage` (v. § 9.3).

5.8 Comandi per le note

Si possono inserire facilmente note a piè di pagina e note a margine.

5.8.1 Note a piè di pagina

Una nota a piè di pagina[2] si ottiene col comando

`\footnote`{*nota*}

[2] Come questa.

LATEX si preoccupa di numerare le note:

```
Una nota a piè di pagina\footnote{Come questa.}
si ottiene...
```

5.8.2 Note a margine

Una nota a margine si ottiene anch'essa molto semplicemente con il comando:

questa è una nota a margine

\marginpar[*testo di sinistra*]{*testo di destra*}

```
Una nota a margine\marginpar{questa è una nota a margine}
si ottiene...
```

Se la nota va posta nel margine sinistro e viene precisato l'argomento opzionale *testo di sinistra*, verrà utilizzato quest'ultimo. Negli altri casi viene utilizzato *testo di destra* e la nota appare nel margine destro[3].

5.9 I riferimenti

Con LATEX, è molto semplice fare riferimento a una parte del documento (pagina, capitolo, tabella, equazione...).

Il comando `\label`{*nome*} permette di piazzare un contrassegno, invisibile in stampa, sulla parte del testo alla quale ci si vuole riferire.

Il comando `\ref`{*nom*} permette poi di fare riferimento alla parte di testo contrassegnata da *nome*. Ovviamente dev'essere qualcosa di numerato, come un capitolo, una sezione, una tabella, un'equazione...

Il comando `\pageref`{*nom*} permette di fare riferimento alla pagina in cui si trova il contrassegno *nome*.

Ad esempio, ho messo un contrassegno su questa sezione come segue:

```
\section{I riferimenti} \label{sez:riferimenti}
```

Posso allora far apparire il numero di questa sezione (5.9, o mi sbaglio?) con il comando:

```
\ref{sez:riferimenti}
```

e alla pagina in cui comincia (23, credo...) con:

```
\pageref{sez:riferimenti}
```

Il parametro del comando `\label` è una combinazione di lettere, cifre e punteggiatura. Personalmente utilizzo una stringa della forma *tipo*`:`*nome* in cui *tipo* rappresenta il tipo del riferimento (`cap` per un capitolo, `sez` per una sezione, `tab` per una tabella, `fig` per una figura...). Per esempio la stringa con la quale ho contrassegnato questa sezione è `sez:riferimenti` perché è una sezione che parla dei riferimenti. Mi sembra un metodo pratico e chiaro.

[3]Nei documenti fronte-retro, i margini sono invertiti nelle pagine pari.

Capitolo 6

Gli ambienti di LaTeX

Gli ambienti sono strutture estremamente potenti di cui si fa un uso intensivo in LaTeX.

Un ambiente è una parte del documento delimitata dai comandi

`\begin`{*ambiente*}

e

`\end`{*ambiente*}

Come tutti i comandi, gli ambienti possono avere delle opzioni e dei parametri, nel qual caso il comando di inizio ambiente assume la forma:

`\begin`{*ambiente*}`[`*opzione*`]`{*parametro*}

Come i comandi, alcuni ambienti hanno delle *varianti asterisco*, che sono delle varianti dell'ambiente:

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

Una dichiarazione (cambiamento di stile o di corpo) effettuata all'interno di un ambiente non si propaga all'esterno. Così:

```
\begin{quote}
Metto il resto del testo in \itshape corsivo. Blabla...
\end{quote}
Ma il testo è normale fuori dall'ambiente.
```

dà:

> Metto il resto del testo in *corsivo. Blabla...*

Ma il testo è normale fuori dall'ambiente.

A questo punto starete utilizzando già da un po' l'ambiente `document`, ma ne esistono molti altri...

6.1 L'ambiente `em`

L'ambiente `em` mette tutto il suo contenuto in stile `\emph`. È così possibile evidenziare uno o più paragrafi:

```
\begin{em}
Il testo contenuto in questo ambiente è messo in evidenza.
\end{em}
```

Che dà luogo a:

Il testo contenuto in questo ambiente è messo in evidenza.

6.2 Gli ambienti `quote` e `quotation`

Questi due ambienti hanno comportamento simile. Entrambi aumentano i margini destro e sinistro del testo che contengono; si differenziano solo per il rientro e la spaziatura dei paragrafi:

- l'ambiente `quotation` mette un rientro all'inizio di ogni paragrafo e una spaziatura normale tra i paragrafi;
- l'ambiente `quote` non mette il rientro ma separa di più i paragrafi tra di loro.

Esempio:

```
Questo è un paragrafo di testo. Non contiene nulla di
speciale ed è fuori da qualsiasi ambiente particolare.
\begin{quotation}
Questo è un esempio di testo su due paragrafi che
serve a mostrare il comportamento degli ambienti di
``citazione''.

Questo è un esempio di testo su due paragrafi che
serve a mostrare il comportamento degli ambienti di
``citazione''.
\end{quotation}
Questo è un paragrafo di testo. Non contiene nulla di
speciale ed è fuori da qualsiasi ambiente particolare.
```

ci dà:
Questo è un paragrafo di testo. Non contiene nulla di speciale ed è fuori da qualsiasi ambiente particolare.

> Questo è un esempio di testo su due paragrafi che serve a mostrare il comportamento degli ambienti di "citazione".
>
> Questo è un esempio di testo su due paragrafi che serve a mostrare il comportamento degli ambienti di "citazione".

Questo è un paragrafo di testo. Non contiene nulla di speciale ed è fuori da qualsiasi ambiente particolare.

Mentre:

```
Questo è un paragrafo di testo. Non contiene nulla di
speciale ed è fuori da qualsiasi ambiente particolare.
\begin{quote}
Questo è un esempio di testo su due paragrafi che
serve a mostrare il comportamento degli ambienti di
``citazione''.

Questo è un esempio di testo su due paragrafi che
serve a mostrare il comportamento degli ambienti di
``citazione''.
\end{quote}
Questo è un paragrafo di testo. Non contiene nulla di
speciale ed è fuori da qualsiasi ambiente particolare.
```

ci dà:

Questo è un paragrafo di testo. Non contiene nulla di speciale ed è fuori da qualsiasi ambiente particolare.

> Questo è un esempio di testo su due paragrafi che serve a mostrare il comportamento degli ambienti di “citazione”.
>
> Questo è un esempio di testo su due paragrafi che serve a mostrare il comportamento degli ambienti di “citazione”.

Questo è un paragrafo di testo. Non contiene nulla di speciale ed è fuori da qualsiasi ambiente particolare.

6.3 L'ambiente `verse`

LATEX mette a disposizione dei poeti l'ambiente `verse`:

- i margini sono aumentati come con l'ambiente `quote`;
- le linee devono finire con `\\` per tornare a capo, tranne l'ultima di ogni strofa;
- le strofe sono separate da linee bianche.

Se LATEX fosse esistito all'epoca di Dante:

```
\begin{verse}
Tanto gentile e tanto onesta pare \\
la donna mia quand'ella altrui saluta, \\
ch'ogne lingua deven tremando muta, \\
```

```
e li occhi no l'ardiscon di guardare.

Ella si va, sententosi laudare, \\
benignamente d'umiltà vestuta; \\
e par che sia una cosa venuta \\
da cielo in terra a miracol mostrare.

Mostrasi sì piacente a chi la mira, \\
che dà per li occhi una dolcezza al core, \\
che 'ntender no la può chi no la prova: \\
e par che de la sua labbia si mova \\
un spirito soave pien d'amore, \\
che va dicendo a l'anima: Sospira.
\end{verse}
```

Tanto gentile e tanto onesta pare
la donna mia quand'ella altrui saluta,
ch'ogne lingua deven tremando muta,
e li occhi no l'ardiscon di guardare.

Ella si va, sententosi laudare,
benignamente d'umiltà vestuta;
e par che sia una cosa venuta
da cielo in terra a miracol mostrare.

Mostrasi sì piacente a chi la mira,
che dà per li occhi una dolcezza al core,
che 'ntender no la può chi no la prova:
e par che de la sua labbia si mova

un spirito soave pien d'amore,
che va dicendo a l'anima: Sospira.

6.4 Gli ambienti `center`, `flushleft` e `flushright`

Questi ambienti permettono di allineare il testo al centro, solo a destra o solo a sinistra. Si può forzare un ritorno a capo con il comando \\.

```
\begin{center}
Ecco un paragrafo centrato. \LaTeX\ effettua
automaticamente l'impaginazione e va a capo
dove è necessario.

Ecco due linee \\
centrate.
\end{center}
```

Ecco un paragrafo centrato. LATEX effettua automaticamente l'impaginazione e va a capo dove è necessario.
Ecco due linee
centrate.

```
\begin{flushleft}
Ecco un paragrafo giustificato a sinistra. \LaTeX\
effettua automaticamente l'impaginazione e va a capo
dove è necessario.

Ecco due linee \\
giustificate a sinistra.
\end{flushleft}
```

Ecco un paragrafo giustificato a sinistra. LATEX effettua automaticamente
l'impaginazione e va a capo dove è necessario.
Ecco due linee
giustificate a sinistra.

```
\begin{flushright}
Ecco un paragrafo giustificato a destra. \LaTeX\
effettua automaticamente l'impaginazione e va a capo
dove è necessario.

Ecco due linee \\
giustificate a destra.
\end{flushright}
```

Ecco un paragrafo giustificato a destra. LATEX effettua automaticamente
l'impaginazione e va a capo dove è necessario.
Ecco due linee
giustificate a destra.

6.5 L'ambiente `verbatim`

L'ambiente `verbatim` è il più semplice: non fa nulla! Si limita a mettere il suo contenuto in stile `\texttt`. I comandi e i caratteri speciali presenti all'interno di quest'ambiente non vengono interpretati.

L'ambiente `verbatim` è utilizzato principalmente per stampare programmi o codice LATEX (tutti gli esempi in questo manuale sono realizzati con l'ambiente `verbatim`).

```
\begin{verbatim}

Non succede niente nell'ambiente verbatim. I comandi
\LaTeX\ e i caratteri speciali (#$%&~_^\{}) non
vengono interpretati.
```

```
\end{verbatim}
```

dà:

```
Non succede niente nell'ambiente verbatim. I comandi
\LaTeX\ e i caratteri speciali (#$%&~_^\{}) non
vengono interpretati.
```

6.6 Gli elenchi: gli ambienti `itemize`, `enumerate` e `description`

Gli ambienti che realizzano elenchi sono usati molto spesso in LATEX. Permettono infatti:

- di areare il testo;
- di migliorarne la leggibilità;
- di strutturare le proprie idee.

La lista qui sopra è stata ottenuta con l'ambiente `itemize` come segue:

```
\begin{itemize}
   \item di areare il testo;
   \item di migliorarne la leggibilità;
   \item di strutturare le proprie idee.
\end{itemize}
```

Ogni elemento della lista comincia con il comando `\item`. Notate il modo in cui ho indentato gli elementi all'interno dell'ambiente `itemize`: in questo modo il sorgente è più leggibile.

L'ambiente `enumerate` si utilizza come `itemize`, ma aggiunge un numero ad ogni elemento:

```
\begin{enumerate}
   \item primo elemento;
   \item secondo elemento;
   \item terzo elemento.
\end{enumerate}
```

dà:

1. primo elemento;
2. secondo elemento;
3. terzo elemento.

L'ambiente `description` permette di scegliere il testo che appare prima di ogni elemento della lista tramite il parametro opzionale del comando `\item`:

```
\begin{description}
   \item[itemize]     per fare liste semplici;
   \item[enumerate]   per fare liste numerate;
   \item[description] per fare liste in cui ogni elemento
      comincia con un testo a piacere.
\end{description}
```

produce:

itemize per fare liste semplici;

enumerate per fare liste numerate;

description per fare liste in cui ogni elemento comincia con un testo a piacere.

Notate che il testo dell'argomento opzionale di `\item` è stato messo automaticamente in grassetto.

Si possono nidificare le liste, cioè mettere delle liste dentro altre liste:

```
Vi consiglio di utilizzare il più possibile gli ambienti per liste
perché:
\begin{enumerate}
   \item sono semplici da usare;
   \item rendono più chiaro il testo:
         \begin{itemize}
            \item areandolo;
            \item facilitandone la lettura;
         \end{itemize}
   \item permettono di strutturare meglio le proprie idee.
\end{enumerate}
```

dà:

> Vi consiglio di utilizzare il più possibile gli ambienti per liste perché:
>
> 1. sono semplici da usare;
> 2. rendono più chiaro il testo:
>   - areandolo;
>   - facilitandone la lettura;
> 3. permettono di strutturare meglio le proprie idee.

Qualche ultima cosa sulle convenzioni tipografiche per gli elenchi:

- per le liste semplici (i cui elementi contengono una sola frase) ogni elemento deve terminare con un punto e virgola, tranne l'ultimo, che termina col punto;

- per le liste i cui elementi contengono due o più frasi, ogni elemento termina con un punto.

6.7 L'ambiente `tabbing`

L'ambiente `tabbing` permette di definire delle *tabulazioni* come su una macchina da scrivere. Il suo uso è assai strano e poco pratico, e quindi vi consiglierei, per quanto possibile, di utilizzare al suo posto l'ambiente `tabular` (v. sez. 6.8).

Ecco un semplice esempio d'uso dell'ambiente `tabbing`:

```
\begin{tabbing}
Seconda linea   \= uffa! \= un po' uno schifo \kill
Prima linea     \> bah   \> un po' uno schifo \\
Seconda linea   \> mah   \> proprio brutto \\
Terza linea     \> uffa! \> terrificante
\end{tabbing}
```

che dà:

Prima linea	bah	un po' uno schifo
Seconda linea	mah	proprio brutto
Terza linea	uffa!	terrificante

Tutto ciò non è molto chiaro... vediamo dunque di spiegare un po':

- le tabulazioni vengono definite con il comando `\=`;
- ci si sposta sulla tabulazione successiva con il comando `\>`;
- il comando `\kill` permette di non stampare la linea sulla quale ci si trova;
- ogni linea (tranne l'ultima e quelle che terminano con `\kill`) devono finire con il comando `\\`.

Nel nostro esempio, abbiamo definito le tabulazioni utilizzando gli elementi più lunghi di ogni colonna (nella prima linea che non è stata stampata a causa del comando `\kill`).

Si possono ridefinire le tabulazioni nel bel mezzo dell'ambiente:

```
\begin{tabbing}
Brigitta 1 \= Filo Sganga 1 \= Paperone 1 \\
Brigitta 2 \> Filo Sganga 2 \> Paperone 2 \\
Gastone 1 \= Paperoga 1 \\
Gastone 2 \> Paperoga 2
\end{tabbing}
```

che ci dà:

Brigitta 1 Filo Sganga 1 Paperone 1
Brigitta 2 Filo Sganga 2 Paperone 2
Gastone 1 Paperoga 1
Gastone 2 Paperoga 2

Esistono altri comandi che permettono di manipolare le tabulazioni, ma non ne parlerò qui. Se veramente volete utilizzare l'ambiente `tabbing`, cf. [1, pagg. 62 e 179].

6.8 Le tabelle: l'ambiente `tabular`

È senz'ombra di dubbio l'ambiente più potente di LATEX, con il quale potrete fare tutte le tabelle possibili ed immaginabili (o quasi).

Cominciamo con un semplice esempio:

```
\begin{tabular}{lcr}
Sparc & SunOS  & 4.1.4 \\
HP    & HP-UX  & 10.20 \\
PC    & NetBSD &   1.2
\end{tabular}
```

dà:

Sparc	SunOS	4.1.4
HP	HP-UX	10.20
PC	NetBSD	1.2

Abbiamo dato in pasto all'ambiente `tabular` un parametro di tre caratteri: ciò significa che la tabella consta di tre colonne. Più in dettaglio:

`l` (***left***) indica che la prima colonna sarà giustificata a sinistra;

`c` (***center***) indica che la seconda colonna sarà centrata;

`r` (***right***) indica che la terza colonna sarà giustificata a destra.

All'interno dell'ambiente `tabular`, le linee terminano con il comando `\\` (tranne l'ultima) e le colonne sono separate da `&`.

Avrete notato che ho messo le `&` in colonna per migliorare la leggibilità della tabella nel sorgente (LATEX se ne infischia, ma voi no) e che ho allineato le colonne nel modo in cui saranno stampate. Trovo che sia una buona idea e vi consiglio di fare altrettanto.

Ecco ora una tabella un po' più complicata:

```
\begin{tabular}{|l|c|r|}
\hline
Sparc & SunOS  & 4.1.4 \\
\hline
HP    & HP-UX  & 10.20 \\
\hline
PC    & NetBSD &   1.2 \\
\hline
\end{tabular}
```

Sparc	SunOS	4.1.4
HP	HP-UX	10.20
PC	NetBSD	1.2

È la stessa di prima, ma ho aggiunto delle linee. Non è difficile:

- le linee verticali si ottengono mettendo dei `|` tra le lettere che definiscono l'allineamento delle colonne nel parmetro dell'ambiente `tabular` (queste linee occupano tutta l'altezza della tabella);
- le linee orizzontali si ottengono con il comando `\hline` tra le linee di testo della tabella (notate che non c'è `\\` dopo `\hline`).

Per capire bene come funziona provate a togliere o aggiungere delle linee.

Ecco un'altra tabella:

```
\begin{tabular}{||p{5cm}||*{2}{c|}|}
\hline
                            & Contenuto  & Quantità \\
\hline
\hline
\bfseries Heineken          &    33 cl   &    10    \\
\hline
\bfseries Guinness          &   1 pinta  &     5    \\
\hline
\bfseries Kronenbourg       &    33 cl   &     0    \\
\hline
\end{tabular}
```

	Contenuto	Quantità
Heineken	33 cl	10
Guinness	1 pinta	5
Kronenbourg	33 cl	0

Qui scopriamo due nuovi parametri dell'ambiente `tabular` :

`p`{***larghezza***} che dichiara una colonna di larghezza *larghezza* (il testo è allineato a sinistra per default, ma possiamo centrarlo con il comando `\centering` o allinearlo a destra con il comando `\raggedright`) ;

`*`{***n***}{***formato***} che è equivalente a n volte la dichiarazione *format* (qui dunque, `c|c|`): utile quindi per migliorare la leggibilità nel caso di tabelle con molte colonne.

Notate che alcune caselle si possono lasciare in bianco, e che una dichiarazione di cambiamento di stile o di corpo in una casella non si ripercuote sulle altre.

Ed eccone un altro:

```
\begin{tabular}{|c|c|c|}
\cline{2-2}
\multicolumn{3}{|c|}{tabella} \\
\multicolumn{3}{|c|}{inutile} \\
\hline
```

```
centrata & centrata & a destra \\
\hline
 Pippo   &  Pippo   & \multicolumn{1}{r|}{Pluto} \\
\hline
\end{tabular}
```

	tabella inutile	
centrata	centrata	a destra
Pippo	Pippo	Pluto

Il comando

`\cline`{*i*-*j*}

traccia una linea orizzontale tra le colonne i e j della tabella.

Si può modificare localmente il formato della tabella con il comando

`\multicolumn`{*numero*}{*formato*}{*testo*}

que sostituisce *numero* colonne della tabella con il testo *testo* conformemente al formato *formato* (guardate la tabella precedente e capirete). Questo comando generalmente è utilizzato per fare in modo che un elemento della tabella occupi più colonne (è questo il caso delle due prime linee della tabella qui sopra) o per modificare l'allineamento di una casella (come l'ultima casella in basso a destra).

Per qualche altra sottigliezza sull'ambiente `tabular`, cf. [1, pagg. 182–185] e [2, capitolo 5].

6.9 Gli ambienti `table` e `figure`

Questi ambienti sono solo delle cornici che permettono di far *galleggiare*[1] tabelle e figure, di dar loro un titolo e numerarle.

È importante poter far "galleggiare" tabelle e figure perché LaTeX non le taglia in fondo alla pagina.

Ecco un esempio tipico di utilizzo dell'ambiente `table` (l'ambiente `figure` si utilizza in modo assolutamente identico):

```
\begin{table}[htbp]
\begin{center}
\begin{tabular}{|l|l|}
\hline
table  & tabelle \\
\hline
figure & disegni \\
```

[1] "*far galleggiare*" significa *spostare se c'è bisogno.*

```
\hline
\end{tabular}
\end{center}
\caption{Esempio d'ambiente \texttt{table}}
\label{tab:table}
\end{table}
```

table	tabelle
figure	disegni

Tabella 6.1: Esempio d'ambiente `table`

Gli argomenti opzionali dell'ambiente `table` sono importanti e determinano l'ordine di preferenza nell'impaginazione:

`h` ***(here)*** qui, se possibile;

`t` ***(top)*** all'inizio di una pagina (questa o la successiva);

`b` ***(bottom)*** alla fine di una pagina (questa o la successiva)

`p` ***(page of floats)*** alla fine del documento.

L'ordine di preferenza per default è `[tbp]`, quindi è preferibile precisare `[htbp]` se si vuole che per quanto possibile la tabella sia stampata laddove appare nel testo.

Abbiamo inoltre utilizzato l'ambiente `center` per centrare la tabella e due comandi:

- il comando `\caption` permette di dare un titolo alla tabella o alla figura;
- il comando `\label`, che deve comparire *dopo* il comando `\caption`, permette di fare riferimento alla tabella tramite i comandi `\ref` o `\pageref`.

Capitolo 7

LaTeX e la matematica

Uno dei punti di forza di LaTeX è sicuramente la facilità di realizzazione di documenti scientifici anche molto complessi. Il modo matematico standard risponde a quasi tutte le esigenze ma in certi casi può essere necessario utilizzare un'estensione specifica, l'estensione amstex (v. [2, cap. 8, pag. 215–257]). Esistono anche moltre altre estensioni, per esempio per strutture molecolari, diagrammi utilizzati in fisica subatomica, circuiti elettrici ecc. (v. [3, cap. 6, pagg. 205–252]).

Le regole tipografiche di scrittura di formule sono quasi tutte rispettate automaticamente da LaTeX; se volete approfondire questo aspetto potete comunque riferirvi a [4, 107–116].

7.1 I modi matematici

LaTeX prevede un gran numero di simboli d'operatori e relazioni e li dispone da solo nel modo appropriato, ma bisogna potergli indicare che stiamo inserendo una formula. Per questo, LaTeX dispone di due modi matematici:

- uno è utilizzato per le formule che devono comparire in mezzo al testo, e corrisponde all'ambiente `math`;
- l'altro è utilizzato per le formule che si desidera far stampare su una linea a sé, ed è l'ambiente `displaymath`.

7.1.1 L'ambiente `math`

È il modo matematico che permette di stampare formule nel corpo del testo:

```
Consideriamo l'uguaglianza \begin{math} x + y + z = n \end{math}.
```

che dà:

Consideriamo l'uguaglianza $x + y + z = n$.

Se fate molto uso di formule matematiche, questa sintassi diventa pesante e diminuisce la leggibilità del sorgente: esistono allora due forme più semplici, `\( ... \)` et `$ ... $`. L'esempio precedente può essere scritto più semplicemente:

```
Consideriamo l'uguaglianza \( x + y + z = n \).
```

oppure:

```
Consideriamo l'uguaglianza $ x + y + z = n $.
```

La forma più usata è naturalmente `$ ... $`.
Notate che gli spazi non hanno alcun effetto in modo matematico, quindi `$x+y+z=n$` darà lo stesso risultato di `$ x + y + z = n $`.

7.1.2 L'ambiente displaymath

È il modo matematico che permette di inserire formule su una linea a sé stante:

```
Ne risulta:
\begin{displaymath}
x + y + z = n
\end{displaymath}
```

Ne risulta:

$$x + y + z = n$$

Così come per l'ambiente `math`, esistono forme più semplici: `\[ ... \]` e `$$ ... $$`. Avremmo potuto scrivere:

```
Ne risulta:
\[
x + y + z = n
\]
```

oppure:

```
Ne risulta:
$$
x + y + z = n
$$
```

ATTENZIONE: non lasciate per nessun motivo una linea bianca all'interno dell'ambiente `displaymath`: otterreste un messaggio d'errore.

Per default, l'ambiente `displaymath` produce formule centrate. Se preferite che siano allineate a sinistra potete utilizzare l'opzione di classe fleqn.

7.2 Strutture di uso frequente in matematica

7.2.1 Indici ed esponenti

Con LATEX, si trovano indici ed esponenti un po' dovunque. Per esempio, i limiti di una sommatoria o di un integrale si inseriscono come se fossero indici ed esponenti.

Un indice si ottiene con il carattere _ ed un esponente con ^:

```
$ x^2 + y^2 = 1 $, $ x_1 = x_2 $, $ x_1^2 = x^2_1 $
```

$x^2 + y^2 = 1$, $x_1 = x_2$, $x_1^2 = x^2_1$

Se l'indice (o l'esponente) consta di più di un carattere bisogna utilizzare delle graffe di raggruppamento:

```
$ x^{2y} = z $, $ a_{ij} = 0 $, $ x^{y^{z}} = e $
```

$x^{2y} = z$, $a_{ij} = 0$, $x^{y^{z}} = e$

Inoltre si possono usare le graffe per indicare a che cosa dev'essere applicato l'indice o l'esponente:

```
Riuscite a vedere la differenza tra $ (x^2+y^2)^n $ e
$ {(x^2+y^2)}^n $?
```

Riuscite a vedere la differenza tra $(x^2+y^2)^n$ e ${(x^2+y^2)}^n$?

Nel primo caso, l'esponente viene applicato al solo carattere che lo precede (la parentesi), e quindi è alla stessa altezza dei quadrati; nel secondo, grazie alle graffe, l'esponente n si applica a tutta l'espressione (x^2+y^2) e quindi viene stampato un po' più in alto, perché le parentesi già contengono due esponenti. Sta a voi preferire una forma o l'altra, visto che entrambe sono corrette dal punto di vista matematico.

7.2.2 Testo all'interno di una formula

Può essere utile mettere qualche parola in una formula, ma per farlo è necessario far tornare LATEX dal modo matematico in modo testo. Questo è possibile con il comando `\mbox`:

```
$$
f(x) > 1 \mbox{ se } x < 3
$$
```

$$f(x) > 1 \mbox{ se } x < 3$$

Notate gli spazi prima e dopo il "se" nel comando `\mbox`: sono indispensabili perché la spaziatura nel modo matematico di LATEX è generata automaticamente e non dipende dagli spazi presenti nel sorgente. Potete rendervene conto da soli:

```
$$
f(x) > 1 \mbox{se} x < 3
$$
```

$$f(x) > 1 \mbox{se} x < 3$$

Naturalmente il comando `\mbox` è inutile con l'ambiente `math`:

```
Quindi $ f(x) > 1 $ se $ x < 3 $.
```

Quindi $f(x) > 1$ se $x < 3$.

7.3 Lettere, simboli e accenti

Nelle formule matematiche si fa spesso uso di lettere greche, accenti e simboli diversi. LaTeX dispone di molti comandi per stamparli.

C'è solo una differenza rispetto al modo testo di LaTeX. In modo matematico non c'è bisogno di utilizzare il comando `\␣` per mettere uno spazio dopo un simbolo, perché la spaziatura viene gestita automaticamente da LaTeX.

7.3.1 Lettere greche

I comandi che permettono di ottenere le lettere greche sono elencati in tab. 7.1.

Minuscole

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

Maiuscole

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

Tabella 7.1: Lettere greche

Le lettere greche minuscole si ottengono con un `\` seguito dal nome della lettera; per le maiuscole basta scrivere in maiuscolo la prima lettera del nome.

Non esiste un comando per ottenere la lettera omicron, basta usare `$o$`.

Per certe lettere minuscole esiste una seconda forma il cui comando comincia per `var`. In particolare vi raccomando di utilizzare `\varepsilon` e `\varphi` al posto di `\epsilon` e `\phi`, perché le prime forme sono più eleganti.

Alcuni comandi (così come tutti quelli che sono presentati in questo capitolo) sono disponibili solo in modo matematico:

```
$\pi$ vale approssimativamente 3,1416.
```

π vale approssimativamente 3,1416.

7.3.2 Operatori

I comandi presentati in tab. 7.2 servono ad ottenere gli operatori.

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

[a] Richiede l'estensione latexsym.

Tabella 7.2: Simboli degli operatori

7.3.3 Relazioni

I comandi in tab. 7.3 generano i simboli delle relazioni.

Si può ottenere la negazione di questi operatori tramite il comando `\not` :

```
$ x \not\in F $
```

$x \not\in F$

7.3.4 Frecce

I comandi che permettono di ottenere delle frecce sono presentati in tab. 7.4.

7.3.5 Altri simboli

Gli altri simboli matematici possono essere ottenuti con i comandi elencati in tab. 7.5.

In più, i comandi `\ldots` e `\cdots` stampano dei puntini di sospensione rispettivamente allineati a sinistra e centrati.

$\leq$	\leq	$\geq$	\geq	$\equiv$	\equiv	$\models$	\models
$\prec$	\prec	$\succ$	\succ	$\sim$	\sim	$\perp$	\perp
$\preceq$	\preceq	$\succeq$	\succeq	$\simeq$	\simeq	$\mid$	\mid
$\ll$	\ll	$\gg$	\gg	$\asymp$	\asymp	$\parallel$	\parallel
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx	$\bowtie$	\bowtie
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong	$\Join$	\Join[a]
$\sqsubset$	\sqsubset[a]	$\sqsupset$	\sqsupset[a]	$\neq$	\neq	$\smile$	\smile
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\doteq$	\doteq	$\frown$	\frown
$\in$	\in	$\ni$	\ni	$\propto$	\propto		
$\vdash$	\vdash	$\dashv$	\dashv				

[a] Richiede l'estensione latexsym.

Tabella 7.3: Simboli delle relazioni

$\leftarrow$	\leftarrow	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\rightarrow$	\rightarrow	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftrightarrow$	\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	$\leadsto$	\leadsto[a]		

[a] Richiede l'estensione latexsym.

Tabella 7.4: Frecce

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

[a] Richiede l'estensione latexsym.

Tabella 7.5: Altri simboli

7.3.6 Accenti

Gli accenti disponibili in modo matematico si ottengono tramite i comandi in tab. 7.6.

$\hat{a}$	\hat{a}	$\acute{a}$	\acute{a}	$\bar{a}$	\bar{a}	$\dot{a}$	\dot{a}
$\check{a}$	\check{a}	$\grave{a}$	\grave{a}	$\vec{a}$	\vec{a}	$\ddot{a}$	\ddot{a}
$\breve{a}$	\breve{a}	$\tilde{a}$	\tilde{a}				

Tabella 7.6: Accenti

Per mettere un accento su un i o un j utilizzate i comandi \imath et \jmath che producono delle lettere senza puntino.

Esistono anche comandi che permettono di avere delle varianti più lunghe di alcuni accenti:

```
$ \widehat{xyz} $, $ \widetilde{abc} $
```

$\widehat{xyz}$, $\widetilde{abc}$

Il comando \overrightarrow fa stampare una freccia:

```
$ \overrightarrow{AB} $
```

$\overrightarrow{AB}$

I comandi \overline e \underline permettono di effettuare una sottolineatura:

```
$ \overline{z+1} = \bar{z} + 1 $, $ \underline{\alpha + \beta} $
```

$\overline{z+1} = \bar{z} + 1$, $\underline{\alpha + \beta}$

Infine, i comandi \overbrace e \underbrace mettono delle graffe al di sopra e al di sotto del testo. Si possono inoltre aggiungere dei commenti alla graffa mettendo un esponente dopo il comando \overbrace o, rispettivamente, un indice dopo il comando \underbrace :

```
$$
a^n = \overbrace{a \times a \times \cdots
\times a}^{n \mbox{ volte}}
$$
```

$$a^n = \overbrace{a \times a \times \cdots \times a}^{n \mbox{ volte}}$$

7.3.7 Sovrapposizione di due simboli

Si possono sovrapporre due simboli grazie al comando:

`\stackrel{`*sopra*`}{`*sotto*`}`

```
$ x \stackrel{f}{\longmapsto} f(x) $
```

$x \stackrel{f}{\longmapsto} f(x)$

Per esempio si può utilizzare il comando `\stackrel` per ottenere dei tensori:

```
$ \stackrel{\Rightarrow}{\sigma} =
  \Lambda ( \stackrel{\Rightarrow}{\varepsilon} ) $
```

$\stackrel{\Rightarrow}{\sigma} = \Lambda (\stackrel{\Rightarrow}{\varepsilon})$

7.4 Modifica della spaziatura

In alcuni casi può essere utile modificare la spaziatura per allontanare o avvicinare alcuni simboli.

`\,`	spazio piccolo	`\!`	spazio negativo piccolo
`\:`	spazio medio	`\quad`	spazio piccolo tra due formule
`\;`	spazio grande	`\qquad`	spazio grande tra due formule

Tabella 7.7: Comandi di spaziatura

Vedremo a tempo debito quando possono servire.

7.5 Modificare lo stile ed il corpo

LaTeX permette di modificare lo stile dei caratteri in modo matematico (v. tab. 7.8).

`$ x + y + 2^ n\Omega\cos\theta $`	$x + y + 2^n\Omega\cos\theta$
`$\mathit{x + y + 2^ n\Omega\cos\theta}$`	$\mathit{x + y + 2^n\Omega\cos\theta}$
`$\mathrm{x + y + 2^ n\Omega\cos\theta}$`	$\mathrm{x + y + 2^n\Omega\cos\theta}$
`$\mathbf{x + y + 2^ n\Omega\cos\theta}$`	$\mathbf{x + y + 2^n\Omega\cos\theta}$
`$\mathsf{x + y + 2^ n\Omega\cos\theta}$`	$\mathsf{x + y + 2^n\Omega\cos\theta}$
`$\mathtt{x + y + 2^ n\Omega\cos\theta}$`	$\mathtt{x + y + 2^n\Omega\cos\theta}$

Tabella 7.8: Comandi di cambiamento di stile

Come potete constatare, i comandi di cambiamento di stile agiscono solo sulle lettere, sui numeri e sulle lettere greche maiuscole.

Il comando `\mathcal` genera delle lettere maiuscole corsive:

```
Sia $\mathcal{C}$ la circonferenza di centro O e raggio 1,
$\mathcal{D}_1$ e $\mathcal{D}_2$ due rette.
```

Sia $\mathcal{C}$ la circonferenza di centro O e raggio 1, $\mathcal{D}_1$ e $\mathcal{D}_2$ due rette.

Ecco che aspetto hanno le maiuscole:

$$\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$$

Come vedrete in seguito, alcuni simboli non hanno la stessa grandezza nell'ambiente `math` e nell'ambiente `displaymath`. Il comando `\displaystyle` permette di obbligare LATEX ad usare, all'interno dell'ambiente `math`, i caratteri e i simboli della stessa grandezza dell'ambiente `displaymath`. Questo comando va usato con moderazione perché può modificare l'interlinea del paragrafo nel caso di formule inserite nel corpo del testo.

Parallelamente, il comando `\textstyle` obbliga LATEX ad utilizzare simboli della stessa grandezza dell'ambiente `math`.

Segnaliamo inoltre i comandi `\scriptstyle` e `\scriptscriptstyle` che rendono la grandezza del testo rispettivamente pari a quella di un esponente e dell'esponente di un esponente (più piccolo di così non si può).

7.6 Frazioni

Le frazioni anche complesse si ottengono semplicemente con il comando:

`\frac{`*numeratore*`}{`*denominatore*`}`

```
$$
\frac{x+1}{x+2}
$$
```

produce:

$$\frac{x+1}{x+2}$$

7.7 Radici

Le radici si ottengono con il comando:

`\sqrt[`*ordine*`]{`*argomento*`}`

```
$ \sqrt{x^2+y^2} $, $ \sqrt[n]{\alpha_i} $
```

dà:

$\sqrt{x^2+y^2}$, $\sqrt[n]{\alpha_i}$

7.8 Delimitatori

I *delimitatori*, presentatati nella tab. 7.9, sono dei simboli che servono a raggruppare delle formule. La loro altezza si adatterà a quella della formula.

<table>
<tr><td>$($</td><td><code>(</code></td><td>$)$</td><td><code>)</code></td><td>$\uparrow$</td><td><code>\uparrow</code></td></tr>
<tr><td>$[$</td><td><code>[</code></td><td>$]$</td><td><code>]</code></td><td>$\downarrow$</td><td><code>\downarrow</code></td></tr>
<tr><td>$\{$</td><td><code>\{</code></td><td>$\}$</td><td><code>\}</code></td><td>$\updownarrow$</td><td><code>\updownarrow</code></td></tr>
<tr><td>$\lfloor$</td><td><code>\lfloor</code></td><td>$\rfloor$</td><td><code>\rfloor</code></td><td>$\Uparrow$</td><td><code>\Uparrow</code></td></tr>
<tr><td>$\lceil$</td><td><code>\lceil</code></td><td>$\rceil$</td><td><code>\rceil</code></td><td>$\Downarrow$</td><td><code>\Downarrow</code></td></tr>
<tr><td>$\langle$</td><td><code>\langle</code></td><td>$\rangle$</td><td><code>\rangle</code></td><td>$\Updownarrow$</td><td><code>\Updownarrow</code></td></tr>
<tr><td>$/$</td><td><code>/</code></td><td>$\backslash$</td><td><code>\backslash</code></td><td></td><td></td></tr>
<tr><td>$\vert$</td><td><code>|</code></td><td>$\Vert$</td><td><code>\|</code></td><td></td><td></td></tr>
</table>

Tabella 7.9: Delimitatori

I delimitatori devono comparire sempre a due a due: uno aperto e uno che chiuso. Un delimitatore aperto si ottiene con il comando `\left` seguito dal delimitatore, e quello chiuso con il comando `\right` seguito dal delimitatore. Ovviamente si possono mettere più coppie di delimitatori una nell'altra.

Se non si vuole far comparire un delimitatore chiuso, bisogna utilizzare il comando `\right.` (`\right` seguito da un punto). Infatti, ad ogni `\left` deve corrispondere un `\right`: il `\right.` non fa stampare nulla, ma serve a chiudere il corrispondente `\left`.

```
$$
\left[
\begin{array}{cc}
a_{11} & a_{12} \\ a_{21} & a_{22}
\end{array}
\right]
$$
```

$$\left[\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array}\right]$$

```
$$
|x| =
\left\{
\begin{array}{rl}
 x & \mbox{si } x \geq 0 \\
-x & \mbox{si } x < 0
\end{array}
\right.
$$
```

$$|x| = \begin{cases} x & \text{si } x \geq 0 \\ -x & \text{si } x < 0 \end{cases}$$

7.9 Simboli a grandezza variabile

I simboli elencati in tab. 7.10 sono a *grandezza variabile*: non hanno cioè la stessa grandezza nei due ambienti `math` e `displaymath`.

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

Tabella 7.10: Simboli a grandezza variabile

Alcuni simboli sono più piccoli in ambiente `math` per poter avere la stessa altezza del resto del testo; inoltre i loro indici ed esponenti non sono disposti nello stesso modo nei due ambienti:

```
$  \sum_{i=0}^{+\infty} x_i \quad \int_a^b f(x) dx  $
$$ \sum_{i=0}^{+\infty} x_i \quad \int_a^b f(x) dx $$
```

$\sum_{i=0}^{+\infty} x_i \quad \int_a^b f(x) dx$

$$\sum_{i=0}^{+\infty} x_i \quad \int_a^b f(x) dx$$

In ambiente `math` si può utilizzare il comando `\displaystyle` che dà ai simboli a grandezza variabile la grandezza che assumerebbero in ambiente `displaymath` (ma in questo modo l'interlinea del testo risulta modificata):

```
Ne risulta che
$ \displaystyle \sum_{n=1}^{+\infty} \frac{1}{n^2} =
  \frac{\pi^2}{6} $.
```

Ne risulta che $\displaystyle \sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$.

Negli intergrali, spesso si usano i comandi di spaziatura per allontanare il dx o per avvicinare i simboli di integrazione degli integrali multipli:

```
$$
\int_a^b f(x) \, dx \qquad
V = \int \!\!\!\! \int \!\!\!\! \int_{\Omega} d\tau
$$
```

$$
\int_a^b f(x) \, dx \qquad V = \iiint_{\Omega} d\tau
$$

Se si usano spesso integrali multipli può essere interessante utilizzare l'estensione amstex (v. [2, cap. 8, pag. 215–257]) che definisce dei comandi specifici per gli integrali multipli (v. [2, pag. 223]).

7.10 Logaritmi e altre funzioni

È d'uso scrivere alcuni nomi di funzioni o operatori in caratteri latini "dritti":

$$
\lim_{x \rightarrow +\infty} \ln x = +\infty
$$

Potremmo scrivere questa uguaglianza così:

```
$$
lim_{x \rightarrow +\infty} ln x = +\infty
$$
```

$$
lim_{x \rightarrow +\infty} ln x = +\infty
$$

Però gli operatori lim e ln non sono dritti. Proviamo allora così:

```
$$
\mbox{lim}_{x \rightarrow +\infty} \mbox{ln} x = +\infty
$$
```

$$
\mbox{lim}_{x \rightarrow +\infty} \mbox{ln} x = +\infty
$$

È già meglio, ma non abbiamo la giusta spaziatura e $x \rightarrow +\infty$ non è sotto il simbolo lim come dovrebbe.

Per risolvere tutti questi problemi LATEX dispone dei comandi elencati in tab. 7.11.

\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

Tabella 7.11: Logaritmi e altri

Il modo giusto di scrivere

$$
\lim_{x \rightarrow +\infty} \ln x = +\infty
$$

è allora:

```
$$
\lim_{x \rightarrow +\infty} \ln x = +\infty
$$
```

Alcuni comandi della tab. 7.11 si comportano come dei simboli a grandezza variabile per quanto riguarda il posizionamento dei loro indici, come ad esempio i limiti:

```
$ \lim_{x \rightarrow 0} \ln x = -\infty $
```

$\lim_{x \rightarrow 0} \ln x = -\infty$

Infine, i comandi `\bmod` e `\pmod` stampano le due forme dell'operatore *modulo*:

```
$ x = a \bmod b $
$ a \equiv b \pmod{n} $
```

$x = a \bmod b$
$a \equiv b \pmod{n}$

7.11 Le tabelle: l'ambiente `array`

Le tabelle (matrici ecc.) si ottengono con l'ambiente `array`, che si utilizza esattamente come `tabular` :

```
$$
{\mathcal A} = \left(
\begin{array}{ccc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}
\right)
$$
```

$${\mathcal A} = \left(
\begin{array}{ccc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}
\right)$$

Il parametro dell'ambiente `array` indica il numero di colonne della tabella ed il loro allineamento. Ogni carattere rappresenta una colonna e può essere:

`l` ***(left)*** la colonna è allineata a sinistra;

`c` ***(center)*** la colonna è centrata;

`r` ***(right)*** la colonna è allineata a destra.

Nel nostro esempio la matrice ha tre colonne (`ccc`), tutte centrate.

All'interno dell'ambiente `array`, le linee sono terminate da `\\` (tranne l'ultima) e le colonne sono separate da `&`.

I comandi `\cdots` ($\cdots$), `\vdots` ($\vdots$) e `\ddots` ($\ddots$) sono usati spesso nelle matrici:

```
$$
{\mathcal A} = \left(
\begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn} \\
\end{array}
\right)
$$
```

$$
{\mathcal A} = \left(
\begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn} \\
\end{array}
\right)
$$

L'ambiente `array` permette di fare ogni tipo di tabella in modo matematico, non solo le matrici:

```
$$
\sum_{\begin{array}{c}
\scriptstyle i>0 \\
\scriptstyle j>0
\end{array}} a_{ij} = \alpha
$$
```

$$
\sum_{\begin{array}{c}
\scriptstyle i>0 \\
\scriptstyle j>0
\end{array}} a_{ij} = \alpha
$$

7.12 Le equazioni numerate: l'ambiente `equation`

L'ambiente `equation` si usa come l'ambiente `displaymath` ma in più numera l'equazione.

```
\begin{equation} \label{eqn:eulero}
e^{i\pi} + 1 = 0
\end{equation}
```

$$e^{i\pi} + 1 = 0 \tag{7.1}$$

Come tutto ciò che è numerato, si può creare un riferimento ad un'equazione col comando `\label`, e poi riferirsi ad essa col comando `\ref`.

Il numero dell'equazione appare a destra per default. Si può fare in modo che compaia a sinistra caricando l'opzione di classe leqno

7.13 Le serie di equazioni: l'ambiente eqnarray

A volte può essere necessario sovrapporre delle formule o svolgere un calcolo allineando più righe su uno stesso simbolo (il segno =, per esempio). Per fare ciò si utilizza l'ambiente `eqnarray` :

```
\begin{eqnarray}
\ln xy     & = & \ln x + \ln y \\
\exp(x+y) & = & \exp x \: \exp y
\end{eqnarray}
```

$$\begin{aligned} \ln xy &= \ln x + \ln y & (7.2) \\ \exp(x+y) &= \exp x \: \exp y & (7.3) \end{aligned}$$

L'ambiente `eqnarray` si usa come una tabella di formato `rcl`.

Si può sopprimere la numerazione di una linea mettendo il comando commande `\nonumber` alla fine della linea:

```
\begin{eqnarray}
\int_1^2 x^2 dx & = & \left[ \frac{x^3}{3} \right]_1^2 \nonumber \\
                & = & \frac{2^3}{3} - \frac{1^3}{3}    \nonumber \\
                & = & \frac{8}{3} - \frac{1}{3}        \nonumber \\
                & = & \frac{7}{3}
\end{eqnarray}
```

$$\begin{aligned} \int_1^2 x^2 dx &= \left[\frac{x^3}{3} \right]_1^2 \\ &= \frac{2^3}{3} - \frac{1^3}{3} \\ &= \frac{8}{3} - \frac{1}{3} \\ &= \frac{7}{3} & (7.4) \end{aligned}$$

Se non si desidera numerare le equazioni è meglio usare la forma stellata dell'ambiente `eqnarray`. L'ambiente `eqnarray*` si usa come l'ambiente `eqnarray`, ma non numera le equazioni:

```
\begin{eqnarray*}
\int_0^\pi \sin x \, dx & = & \left[ - \cos x \right]_0^\pi \\
                        & = & - \cos \pi + \cos 0           \\
                        & = & 1 + 1                         \\
                        & = & 2
\end{eqnarray*}
```

$$\begin{aligned}\int_0^\pi \sin x \, dx &= \left[- \cos x \right]_0^\pi \\ &= - \cos \pi + \cos 0 \\ &= 1 + 1 \\ &= 2\end{aligned}$$

7.14 Teoremi, definizioni...

Teorema 1 (di Babafou) *LATEX non si limita a formule e segni strani, permette anche di fare dei bellissimi teoremi (come questo).*

Il comando

`\newtheorem`{*nome*}{*testo*}

definisce un nuovo ambiente di nome *nome*.

Per esempio il teorema di Babafou che compare qui sopra è stato realizzato così:

```
\newtheorem{teorema}{Teorema}

\begin{teorema}[di Babafou]
\LaTeX\ non si limita a formule e segni strani,
permette anche di fare dei bellissimi teoremi
(come questo).
\end{teorema}
```

L'ambiente creato dal comando `\newtheorem` compie le seguenti azioni:

1. stampa *testo* (secondo parametro del comando `\newtheorem`) ;
2. poi un numero (ad ogni ambiente creato viene associata una numerazione distinta);
3. poi eventualmente il parametro opzionale (per esempio l'autore del teorema);
4. ed infine il suo contenuto, in corsivo.

Si possono creare numerosi ambienti diversi, uno per i teoremi, uno per le definizioni, uno per i lemmi, uno per le ipotesi, ognuno con un contatore distinto.

Capitolo 8

Bibliografia ed indice analitico

Realizzare una bibliografia o un indice analitico generalmente è assai complicato. LaTeX utilizza due altri programmi, BibTeX e *MakeIndex*, per rendere questo lavoro semplice ed efficace.

8.1 Fare una bibliografia

In LaTeX, ci sono due modi di fare una bibliografia:

- l'ambiente `thebibliography` ;
- il programma BibTeX.

8.1.1 L'ambiente `thebibliography`

Ecco un esempio di bibliografia realizzata con l'ambiente `thebibliography`:

```
\clearpage

\addcontentsline{toc}{chapter}{Bibliografia}

\begin{thebibliography}{9}
   \bibitem{lamport-latex} Leslie Lamport.
                           \emph{\LaTeX: a document
                                 preparation system}.
                           Addison-Wesley, 1994.
   \bibitem{companion}     Michel Goossens,
                           Frank Mittelbach,
                           and Alexander Samarin.
                           \emph{The \LaTeX\ Companion}.
                           Addison-Wesley, 1994.
\end{thebibliography}
```

Che produce:

[1] Leslie Lamport. *LaTeX: a document preparation system.* Addison-Wesley, 1994.

[2] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, 1994.

Il comando `\clearpage` forza la stampa delle tabelle e figure che non sono ancora state stampate, e permette di cominciare su una pagina bianca.

Successivamente, si aggiunge il titolo “bibliografia” all’indice (`toc`). Il secondo argomento del comando `\addcontentsline` precisa il livello del titolo da aggiungere, `chapter` per la classe report, `section` per la classe article.

L’ambiente `thebibliography` si usa in modo simile all’ambiente `itemize`. Ogni elemento della bibliografia comincia con il comando `\bibitem`, che prende come parametro la stringa che identifica, come con `\label`, il documento in questione. Di seguito mettiamo l’autore dell’opera, il titolo (in corsivo), l’editore e l’anno di pubblicazione.

Ogni elemento della bibliografia viene contrassegnato da un numero tra parentesi quadre. È possibile assegnare un’etichetta personalizzata col parametro opzionale di `\bibitem`:

```
\bibitem[Lamport 94]{lamport-latex} Leslie Lamport, ...
```

Che produce:

[Lamport 94] Leslie Lamport, ...

Il parametro dell’ambiente `thebibliography` precisa la lunghezza massima di queste etichette:

- se non si usano etichette personalizzate generalmente si mette 9 nel caso in cui ci siano meno di dieci opere in bibliografia, 99 se ne abbiamo meno di cento ecc.;
- se si usano etichette personalizzate, si mette l’etichetta più lunga.

Il comando `\cite` viene usato per fare riferimento ad un elemento della bibliografia, e stampa il numero o l’etichetta dell’opera:

```
Cf.~\cite{lamport-latex} per maggiori dettagli.
```

Cf. [1] per maggiori dettagli.

Si possono indicare capitoli, pagine, ecc. nel riferimento tramite il parametro opzionale di `\cite` :

```
Cf.~\cite[page 187]{lamport-latex} per
maggiori dettagli.
```

Cf. [1, page 187] per maggiori dettagli.

8.1.2 BibTEX

L'ambiente `thebibliography` ha tutte le funzionalità di cui c'è bisogno per una bibliografia, ma anche alcuni inconvenienti:

- ci obbliga a rifare la bibliografia per ogni documento, anche se c'è solo qualche opera in più;
- in caso di aggiornamento di un documento, bisogna modificare le bibliografie di tutti i documenti in cui è citato.

Da qui l'idea di creare una base di dati delle opere piuttosto che avere una bibliografia indipendente per ogni documento, e di generare automaticamente la bibliografia a partire dalla base di dati, in funzione delle citazioni.

La base di dati dei riferimenti bibliografici

Una base di dati dei riferimenti bibliografici è un file di estensione `.bib` che contiene un certo numero di record come questo:

```
@BOOK
{
   lamport-latex,
   AUTHOR    = "Leslie Lamport",
   TITLE     = "{\LaTeX}: a document preparation system",
   PUBLISHER = "Addison-Wesley",
   YEAR      = "1994",
   EDITION   = "Second"
}
```

Ogni record corrisponde ad un'opera il cui tipo è indicato per primo, subito dopo il carattere `@`. Si tratta in questo caso di un libro, quindi di tipo `BOOK`.

Successivamente indichiamo una chiave che servirà a identificare l'opera nelle vostre basi di dati e a citarla nei documenti come parametro del comando `\cite`. Il formato della chiave è libero ma personalmente uso il nome dell'autore seguito da `:` ed da una parola che caratterizza l'opera.

ATTENZIONE: le chiavi devono essere uniche nelle basi di dati.

Poi si riempie una serie di campi che definiscono l'opera (autore, titolo, ecc.). Per un dato tipo d'opera, alcuni campi sono obbligatori ed altri facoltativi. Ogni campo assume la forma

`NOME_DEL_CAMPO = "`*contenuto del campo*`"`

I campi sono separati tra di loro e dalla chiave con delle virgole.

I diversi tipi di record

Ogni record contiene uno o più campi. Un campo può essere:

obbligatorio
Questo campo è indispensabile affinché BibTEX generi la bibliografia.

opzionale
Questo campo non è indispensabile ma verrà utilizzato, se specificato, per completare le informazioni relative all'opera in questione.

ignorato
Questo campo è inutile, il suo contenuto verrà ignorato da BibTEX.

Ecco la lista dei tipi di record riconosciuti da BibTEX. Per ogni tipo sono indicati i campi obbligatori e opzionali (tutti gli altri sono ignorati) il cui significato verrà spiegato nel paragrafo successivo.

`ARTICLE`
Articolo apparso in un giornale o rivista.
Campi obbligatori: `AUTHOR`, `TITLE`, `JOURNAL`, `YEAR`.
Campi opzionali: `VOLUME`, `NUMBER`, `PAGES`, `MONTH`, `NOTE`.

`BOOK`
Libro regolarmente pubblicato da una casa editrice.
Campi obbligatori: `AUTHOR` o `EDITOR`, `TITLE`, `PUBLISHER`, `YEAR`.
Campi opzionali: `VOLUME` o `NUMBER`, `SERIES`, `ADDRESS`, `EDITION`, `MONTH`, `NOTE`.

`BOOKLET`
Libro che viene distribuito senza un editore.
Campo obbligatorio: `TITLE`.
Campi opzionali: `AUTHOR`, `HOWPUBLISHED`, `ADDRESS`, `MONTH`, `YEAR`, `NOTE`.

`CONFERENCE`
Identico a `INPROCEEDINGS`.

`INBOOK`
Parte di un libro, generalmente senza titolo: un capitolo, un paragrafo o qualche pagina.
Campi obbligatori: `AUTHOR` o `EDITOR`, `TITLE`, `CHAPTER` e/o `PAGES`, `PUBLISHER`, `YEAR`.
Campi opzionali: `VOLUME` o `NUMBER`, `SERIES`, `TYPE`, `ADDRESS`, `EDITION`, `MONTH`, `NOTE`.

`INCOLLECTION`
Parte di un libro con un titolo proprio.
Campi obbligatori: `AUTHOR`, `TITLE`, `BOOKTITLE`, `PUBLISHER`, `YEAR`.
Campi opzionali: `EDITOR`, `VOLUME` o `NUMBER`, `SERIES`, `TYPE`, `CHAPTER`, `PAGES`, `ADDRESS`, `EDITION`, `MONTH`, `NOTE`.

`INPROCEEDINGS`
Articolo nei lavori di una conferenza.
Campi obbligatori: `AUTHOR`, `TITLE`, `BOOKTITLE`, `YEAR`.
Campi opzionali: `EDITOR`, `VOLUME` o `NUMBER`, `SERIES`, `PAGES`, `ADDRESS`, `MONTH`, `ORGANIZATION`, `PUBLISHER`, `NOTE`.

`MANUAL`
Documentazione tecnica.
Champ obligatoire : `TITLE`.
Campi opzionali: `AUTHOR`, `ORGANIZATION`, `ADDRESS`, `EDITION`, `MONTH`, `YEAR`, `NOTE`.

`MASTERSTHESIS`
Tesi di laurea.
Campi obbligatori: `AUTHOR`, `TITLE`, `SCHOOL`, `YEAR`.
Campi opzionali: `TYPE`, `ADDRESS`, `MONTH`, `NOTE`.

`MISC`
Tipo da utilizzare quando nessuno degli altri si rivela appropriato.
Campi obbligatori: nessuno.
Campi opzionali: `AUTHOR`, `TITLE`, `HOWPUBLISHED`, `MONTH`, `YEAR`, `NOTE`.

`PHDTHESIS`
Tesi di dottorato.
Campi obbligatori: `AUTHOR`, `TITLE`, `SCHOOL`, `YEAR`.
Campi opzionali: `TYPE`, `ADDRESS`, `MONTH`, `NOTE`.

`PROCEEDINGS`
Lavori di una conferenza.
Campi obbligatori: `TITLE`, `YEAR`.
Campi opzionali: `EDITOR`, `VOLUME` o `NUMBER`, `SERIES`, `ADDRESS`, `MONTH`, `ORGANIZATION`, `PUBLISHER`, `NOTE`.

`TECHREPORT`
Rapporto pubblicato da un'università, scuola, istituto o altra istituzione.
Campi obbligatori: `AUTHOR`, `TITLE`, `INSTITUTION`, `YEAR`.
Campi opzionali: `TYPE`, `NUMBER`, `ADDRESS`, `MONTH`, `NOTE`.

`UNPUBLISHED`
Documento con un autore ed un titolo, ma che non è mai stato pubblicato.
Campi obbligatori: `AUTHOR`, `TITLE`, `NOTE`.
Campi opzionali: `MONTH`, `YEAR`.

I diversi tipi di campo

`ADDRESS`
Indirizzo dell'editore (`PUBLISHER`) o dell'istituzione (`INSTITUTION`).

`ANNOTE`
: Annotazione.

`AUTHOR`
: Nome dell'autore (o degli autori, nel formato descritto più avanti).

`BOOKTITLE`
: Titolo di una parte di un libro o di un articolo nei lavori di una conferenza.

`CHAPTER`
: Numero del capitolo (o di una qualunque parte del documento).

`EDITION`
: Edizioni di un libro.

`EDITOR`
: Nome del redattore.

`HOWPUBLISHED`
: Tipo di pubblicazione.

`INSTITUTION`
: Istituzione editrice di un rapporto.

`JOURNAL`
: Nome di un giornale.

`MONTH`
: Mese di pubblicazione dell'opera.

`NOTE`
: Informazioni supplementari che possono aiutare il lettore.

`NUMBER`
: Numero di un giornale o rivista.

`ORGANIZATION`
: Organizzazione di patrocinio della conferenza o che pubblica il giornale.

`PAGES`
: Uno o più numeri di pagina.

`PUBLISHER`
: Nome dell'editore.

`SCHOOL`
: Nome della scuola o dell'università nella quale è stata realizzata la tesi.

`SERIES`
: Nome di una collezione di volumi.

`TITLE`
Titolo dell'opera.

`TYPE`
Tipo di un rapporto tecnico.

`VOLUME`
Volume di un giornale o di un libro.

`YEAR`
Anno di pubblicazione.

Alcune precisazioni

Per alcuni tipi di record (lavori di conferenze, articoli), BIBTEXmette automaticamente in minuscolo il titolo dell'opera. Ciò può risultare fastidioso nel caso di acronimi, sigle ecc. che si vogliono in maiuscolo. In questo caso, è sufficiente mettere tra parentesi graffe la o le parole che non si vogliono in minuscolo:

```
TITLE = "{TCP-IP} e lo Zen di {Confucio}"
```

Se un campo contiene un comando LATEX, il comando dev'essere messo tra parentesi graffe per non confondere BIBTEX:

```
TITLE = "{\LaTeX}: a document preparation system"
```

Quando un campo `AUTHOR` o `EDITOR` contiene più nomi, i nomi devono essere separati tra loro dalla parola `and`.

Generare la bibliografia con BibTEX

Una volta che avete creato le vostre basi di dati, il più è fatto.

Ora, quando volete fare riferimento ad un'opera, usate il comando `\cite` mettendo come parametro la chiave corrispondente:

```
Cf.~\cite{lamport-latex} per maggiori dettagli.
```

Bisogna altresì indicare a LATEX dove mettere la bibliografia tramite il comando `\bibliography`. Questo comando prende come parametro una lista di file contenenti le basi di dati (ma senza estesione `.bib`):

```
\bibliography{file1,file2,file3}
```

Naturalmente, in questi file dovranno trovarsi tutte le opere citate nel documento.

Bisogna anche indicare a BIBTEX lo stile da utilizzare per impaginare la bibliografia con il comando `\bibliographystyle`. Il parametro di questo comando è lo stile da utilizzare.

Gli stili standard di bibliografia di LATEX sono :

`plain`
: Opere in ordine alfabetico, contrassegnate da numeri.

`unsrt`
: Opere in ordine di citazione, contrassegnate da numeri.

`alpha`
: Opere contrassegnate da etichette del tipo "Lam94" formate a partire dal nome dell'autore e dall'anno di pubblicazione.

`abbrv`
: Come lo stile `plain`, ma i nomi di battesimo, i nomi dei mesi e dei giornali sono abbreviati.

Esistono molti altri stili di bibliografia.

Anche in questo caso bisognerà includere nell'indice la linea relativa alla bibliografia con il comando `\addcontentsline`, perché non viene fatto automaticamente né da LaTeX né da BibTeX:

```
\addcontentsline{toc}{chapter}{Bibliografia}
```

Riassumendo, a parte i comandi `\cite` disseminati nel documento, l'inclusione della bibliografia si fa così:

```
\addcontentsline{toc}{chapter}{Bibliografia}
\bibliographystyle{unsrt}
\bibliography{file1,file2,file3}
```

Se volete includere in bibliografia delle opere non citate nel documento utilizzate il comando `\nocite` (messo ad esempio subito prima del comando `\bibliographystyle`), con la lista delle chiavi delle opere, separata da virgole, come parametro:

```
\nocite{lamport-latex,companion}
```

È anche possibile includere nella bibliografia tutte le opere contenute nella base di dati mettendo `*` come parametro di `\nocite`:

```
\nocite{*}
```

Infine, per generare la bibliografia ed includerla nel documento, bisogna:

- Compilare il documento con LaTeX:

  ```
  latex documento.tex
  ```

- Eseguire BibTeX sul documento:

  ```
  bibtex documento
  ```

(non mettete l'estensione `.tex`)
BibTeX crea un file `documento.bbl`, che contiene la bibliografia impaginata secondo lo stile indicato dal comando `\bibliographystyle`.

- Ricompilate due volte con LaTeX per includere la bibliografia e aggiornare tutti i riferimenti.

Per maggiori informazioni a riguardo di BibTeX, consultate [5], [1, pagg. 69–71 e 155–164] e [2, cap. 13].

8.2 Creare un indice analitico

La realizzazione di un indice analitico è molto semplice con LaTeX. Basta inserire nel testo dei comandi `\index` per generare i numeri di pagina.

Ma prima bisogna modificare leggermente il documento:

- caricate l'estensione makeidx;
- aggiungete il comando `\makeindex` nel preambolo;
- mettete il comando `\printindex` laddove volete far comparire l'indice analitico (generalmente alla fine del documento, subito prima del comando `\end{document}`).

Per un documento con indice analitico, la sequenza di compilazione è la seguente:

- la prima compilazione genera l'indice e lo mette in un file `.idx` (è a questo che serve il comando `\makeindex`);
- il programma *MakeIndex* formatta l'indice e lo mette in un file `.ind`;
- la seconda compilazione include l'indice analitico e genera l'indice (sommario);
- la terza compilazione include definitivamente l'indice nel file.

L'uso di *MakeIndex* è dettagliato nella § 8.2.2 ed il modo di effettuare una compilazione automatica tramite un file `Makefile` nella § 8.3. Inoltre potete consultare [6] per una descrizione più completa della realizzazione di un indice analitico con LaTeX.

8.2.1 Il comando `\index`

Per generare una voce nell'indice, utilizzate il comando `\index` :

```
La parola workstation\index{workstation} è nell'indice.
```

Tutto questo stampa:

> La parola workstation è nell'indice.

e genera una voce per "workstation" nell'indice analitico.

Si può mettere tutta un'espressione nell'indice:

```
È una gran figata\index{gran figata} mettere tutta
un'espressione nell'indice.
```

Per generare delle sotto-voci bisogna separare la voce principale e la sotto-voce con il carattere `!`:

```
Qui parliamo delle sotto-voci
dell'indice\index{indice!sotto-voci}
```

Infine si può mettere del testo alla posizione alfabetica identificata da un altro testo con il carattere `@`:

```
Metto il simbolo $\alpha$\index{alpha@$\alpha$}
nell'indice al posto di ``alpha'' nell'ordine
alfabetico.
```

A volte può rivelarsi utile l'uso delle graffe di raggruppamento:

```
blablabla corsivo\index{corsivo@\textit{corsivo}}
```

8.2.2 Formattazione dell'indice analitico: il programma *MakeIndex*

Una volta generato l'indice analitico (nel file `.idx`), bisogna impaginarlo con *MakeIndex*. Questo programma crea un file `.ind` che può allora essere incluso nel documento con il comando `\printindex`. *MakeIndex* viene richiameto con la linea:

```
makeindex pippo.idx
```

Il formato utilizzato per default da *MakeIndex* è abbastanza brutto. Vi consiglio di usare il file di formato `index.ist` che trovate qui di seguito:

```
headings_flag           1
heading_prefix          "\n  \\item \\textbf{"
heading_suffix          "}"
symhead_positive        "Simboli"
symhead_negative        "simboli"
numhead_positive        "Numeri"
numhead_negative        "numeri"
delim_0                 " \\dotfill\\ "
delim_1                 " \\dotfill\\ "
delim_2                 " \\dotfill\\ "
```

Ora richiamate *MakeIndex* così:

```
makeindex -s index.ist pippo.idx
```

E l'indice sarà formattato come quello di questo manuale.

8.3 Compilazione automatica tramite un file `Makefile`

È possibile automatizzare la compilazione utilizzando un file `Makefile`. Per un file `pippo.tex`, il `Makefile` seguente esegue nell'ordine:

- la compilazione e generazione dell'indice analitico;
- la sua formattazione;
- le due compilazioni di generazione e inclusione dell'indice;
- la conversione del documento in formato PostScript.

```
pippo.ps : pippo.dvi
        dvips pippo.dvi

pippo.dvi : pippo.tex pippo.ind
        latex pippo.tex
        latex pippo.tex

pippo.ind : pippo.idx
        latex pippo.tex
        makeindex -s index.ist pippo.idx

clean :
        rm -f pippo.aux pippo.idx pippo.ilg pippo.ind \
              pippo.lof pippo.log pippo.lot pippo.toc
```

Basta battere `make` e aspettare un po' per avere un documento pronto per la stampa. Tra l'altro, `make clean` permette di cancellare tutti i file intermediari generati da LaTeX, tranne `pippo.tex` e `pippo.ps`.

Capitolo 9

LaTeX da professionisti

Questo capitolo spiega come realizzare con LaTeX delle funzioni per così dire "complesse"... ma non in LaTeX!

9.1 Definizione di nuovi comandi e nuovi ambienti

9.1.1 Definizione di nuovi comandi

Si possono definire comandi personalizzati grazie al comando

`\newcommand{`*nome*`}[`*numero di parametri*`]{`*definizione*`}`

In soldoni, ogni volta che LaTeX troverà il comando `\nome`, lo sostituirà con *definizione*.

Le definizioni dei comandi possono trovarsi nel preambolo o nel corpo del documento.

Ecco un semplice esempio:

```
\newcommand{\mi}{Masterando Istituzionale}
```

Ora posso utilizzare il comando `\mi` che stamperà "Masterando Istituzionale".

```
Un master non è un master senza il \emph{\mi}.
```

Un master non è un master senza il *Masterando Istituzionale*.

Si possono anche creare dei comandi con dei parametri. Basterà precisarne il numero nel parametro *numero di parametri* di `\newcommand` :

```
\newcommand{\itfootnote}[1]{\footnote{\textit{#1}}}
```

Il comando `\itfootnote` crea una nota a piè di pagina in corsivo[1]. È un comando che ha bisogno di un parametro (qui, *numero di parametri* vale 1). Nella definizione del comando, si fa poi riferimento al primo parametro con `#1` (se ce ne fosse stato un secondo, `#2`, ecc.).

[1] *E funziona, come vedete: si possono anche nidificare comandi ovviamente, tutto questo grazie al Masterando Istituzionale.*

9.1.2 Definire nuovi ambienti

È anche possibile definire nuovi ambienti con il comando

`\newenvironment`{*nome*}`[`*numero di parametri*`]`{*definizione di apertura*}{*definizione di chiusura*}

Le definizioni di ambienti possono trovarsi nel preambolo o nel corpo del documento.

Quando LATEX incontrerà `\begin{nom}`, lo sostituirà con *definizione di apertura* e quando incontrerà `\end{nom}`, lo sostituirà con *definizione di chiusura*:

```
\newenvironment{italitemize}{\begin{itemize} \itshape}{\end{itemize}}

\begin{italitemize}
   \item beh,
   \item pare che funzioni!
\end{italitemize}
```

- *beh,*
- *pare che funzioni!*

Generalmente i nuovi ambienti vengono creati a partire di ambienti esistenti (come qui).

È possibile creare degli ambienti con argomento. Se ne precisa il numero con *numero di argomenti* in `\newenvironment` e si utilizzano nella *definizione di apertura* con `#1`, `#2`...

```
\newenvironment{tab}[1]%
{\begin{tabular}{|#1|}\hline}%
{\hline\end{tabular}}

\begin{tab}{cc}
     prova      & dell' \\
    ambiente    & tab \\
\end{tab}
```

prova	dell'
ambiente	tab

Notate in che modo ho utilizzato il comando `\newenvironment`: ho messo le definizioni di apertura e chiusura del nuovo ambiente ognuna su una linea a sé stante (è più chiaro). Per questo, però, ho dovuto mettere un`%` alla fine delle due prime linee. Questo `%` indica un commento, ma non è seguito da testo. A cosa serve? Serve ad impedire che LATEX consideri il ritorno a capo come uno spazio. In questo modo posso mettere la definizione del nuovo ambiente su più linee. Se non avessi messo il `%`, LATEX avrebbe visto degli spazi tra gli ultimi tre parametri di `\newenvironment`, e avrebbe segnalato un errore di sintassi.

9.2 Includere delle immagini PostScript

In un documento LATEX è possibile includere delle immagini in formato *Encapsuled PostScript* generate con qualsiasi strumento (*MatLab*, *xvgr*, *xfig*...). Questo formato è una variante del PostScript per stampanti che omette tutte le informazioni relative all'impaginazione (tipo di carta, margini ecc.) ed è quindi adatto all'inclusione in un documento.

Per poter includere questi file in un documento bisogna innanzitutto caricare l'estensione graphicx.

Nel punto in cui volete il disegno usate poi il comando

`\includegraphics{`*disegno.eps*`}`

È preferibile mettere il disegno in una cornice (`figure`) per potergli mettere una legenda ed un numero, e soprattutto per poterlo spostare se rischia di essere tagliato a fondo pagina.

È anche buona pratica centrarlo con l'ambiente `center`.

Possiamo anche specificare la grandezza che l'immagine deve assumere, in larghezza:

`\includegraphics[width=`*larghezza*`]{`*disegno.eps*`}`

o in altezza:

`\includegraphics[height=`*altezza*`]{`*disegno.eps*`}`

Ecco quindi il codice tipicamente utilizzato per includere un disegno in un documento LATEX:

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

\usepackage[latin1]{inputenc}
\usepackage[italian]{babel}
\usepackage{graphicx}

% ...

\begin{figure}[htbp]
\begin{center}
\includegraphics[width=10cm]{disegno.eps}
\end{center}
\caption{Disegno}
\label{fig:dessin}
\end{figure}
```

9.3 Creare un frontespizio personalizzato: l'ambiente `titlepage`

Avrete senz'altro notato che il comando `\maketitle` genera un frontespizio molto scarno. È un peccato che un programma di alte potenzialità come LATEX debba accontentarsi di copertine così sobrie.

Fortunatamente è possibile creare frontespizi presonalizzati grazi all'ambiente `titlepage`. All'interno, siete completamente padroni dell'impaginazione. Generalmente si utilizza:

- l'ambiente `center`, per centrare il testo;
- i comandi di cambiamento di corpo (`\large`...), per scrivere il titolo in grande;
- i comandi di spaziatura `\vspace` e `\hspace` per areare il testo;
- si può anche includere un'immagine con `\includegraphics` (come si è fatto per il logo dell'ENSTA) ;

L'unico vincolo è che l'ambiente `titlepage` deve trovarsi subito dopo il comando `\begin{document}` e che il frontespizio deve stare tutto in una sola pagina (beh... mi sembra naturale).

Appendice A

Principali regole tipografiche

Le regole tipografiche sono diverse a seconda delle lingue, soprattutto per quanto riguarda la punteggiatura.

A.1 Regole tipografiche italiane

Le regole tipografiche da seguire nello scrivere un testo in italiano sono le seguenti:

- punto, virgola, punto e virgola, due punti, punto esclamativo e punto interrogativo sono *attaccate* alla parola che li precede, mentre sono *separate* con uno spazio dalla parola che li segue;
- parentesi (di ogni tipo), virgolette e trattini sono sempre *attaccate* al testo che delimitano, e *separate* con uno spazio dal resto del testo (a meno che non siano seguite da punto, virgola, punto e virgola, due punti, punto esclamativo o punto interrogativo, nel qual caso questi segni di interpunzione sono attaccati alla parentesi, virgoletta o trattino chiuso);
- i puntini di sospensione si scrivono attaccati alla parola che li precede;
- gli accenti in italiano sono sempre *gravi* (`), tranne nelle parole accentate che finiscono in -ché (*benché, perché, finché* ecc.), in *né* e *sé*;
- i puntini di sospensione devono essere più spaziati tra loro che tre punti normali uno dopo l'altro: per questo, LaTeX dispone del comando `\ldots` per generare i tre puntini (non dimenticate il comando `\␣` per fare lo spazio dopo i puntini di sospensione e non dimenticate di separarli dalla parola precedente con uno spazio insecabile).

In LaTeX, le virgolette vanno scritte `` et '' (e non "). Il risultato è “questo”.

Come esempio di applicazione delle regole tipografiche, ho modificato leggermente la punteggiatura del testo che segue:

```
Partimmo un lunedì (in direzione sud) cercando
l'altra strada che ci ha portato qui. Vedemmo
Pancho Villa e la ``rivoluzione''; ci disse:
--- Non è morta, ci sono nuove idee! --- Siamo
umani, solo umani: non può andare così. E poi
Toro Seduto (incazzato anche con noi)~\ldots\ mi scusai
per la storia ma non generalizzare, le droghe,
il gioco e il resto non vengono dal cielo. Lo
so, non è progresso, ma è un'orgia di idiozia?
```

In stampa otteniamo:

> Partimmo un lunedì (in direzione sud) cercando l'altra strada che ci ha portato qui. Vedemmo Pancho Villa e la "rivoluzione"; ci disse: — Non è morta, ci sono nuove idee! — Siamo umani, solo umani: non può andare così. E poi Toro Seduto (incazzato anche con noi) ... mi scusai per la storia ma non generalizzare, le droghe, il gioco e il resto non vengono dal cielo. Lo so, non è progresso, ma è un'orgia di idiozia?

A.2 Regole tipografiche inglesi

In inglese le regole tipografiche sono sostanzialmente le stesse che in italiano. Ecco quindi un esempio anglosassone che vi suonerà familiare...

```
We all came out to Montreux, on the Lake Geneva
shoreline, to make records with a mobile: we didn't
have much time.  Frank Zappa and the Mothers were at
the best place around; but some stupid with a flare
gun burned the place to the ground \ldots\ Smoke on
the water! Fire in the sky? Smoke on the water!
```

Otteniamo:

> We all came out to Montreux, on the Lake Geneva shoreline, to make records with a mobile: we didn't have much time. Frank Zappa and the Mothers were at the best place around; but some stupid with a flare gun burned the place to the ground ... Smoke on the water! Fire in the sky? Smoke on the water!

A.3 Regole tipografiche francesi

Per i cultori della lingua di Proust le cose sono un po' più difficili. Le regole da seguire sono queste:

- la punteggiatura semplice (punto, virgola, puntini di sospensione) si scrive come in italiano, cioè *attaccata* alla parola precedente e *separata* da quella seguente con uno spazio;

- la punteggiatura doppia (due punti, punto e virgola, punto interrogativo e punto esclamativo) dev'essere *separata* con uno spazio sia dalla parola precedente che da quella seguente;
- parentesi e trattini seguono le stesse regole dell'italiano;
- i puntini si attaccano alla parola precedente e si scrivono `...` invece che `ldots`;
- le virgolette devono essere "alla francese" (vengono chiamate « guillemets ») e si scrivono nel sorgente `<<` e `>>`; si separano con uno spazio dal testo che precede e segue e con uno spazio insecabile dal testo incluso nelle virgolette.

Complicato? Date un'occhiata all'esempio "telefonico" seguente:

```
J'avais un ami, mais il est parti. Ce sens à ma vie,
il n'est plus en vie... Il m'a tout donné, puis s'est
effacé, sans me déranger ; et je crois j'ai pleuré,
j'ai pleuré. Tout au fond, l'air d'un con : envie de
me casser. Voudrais bien essayer, voudrais bien
continuer, continuer ! Un moment, un instant, j'ai cru
oublier. <<~En parlant, en marchant, à Paris, à minuit,
dans ma ville ?~>>
```

e otteniamo:

> J'avais un ami, mais il est parti. Ce sens à ma vie, il n'est plus en vie... Il m'a tout donné, puis s'est effacé, sans me déranger ; et je crois j'ai pleuré, j'ai pleuré. Tout au fond, l'air d'un con : envie de me casser. Voudrais bien essayer, voudrais bien continuer, continuer ! Un moment, un instant, j'ai cru oublier. « En parlant, en marchant, à Paris, à minuit, dans ma ville ? »

Bibliografia

[1] Leslie Lamport. *LaTeX: a document preparation system.* Addison-Wesley, Reading, Massachusetts, second edition, 1994.

[2] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1994.

[3] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX Graphics Companion.* Addison-Wesley, Reading, Massachusetts, 1997.

[4] Imprimerie nationale. *Lexique des règles typographiques en usage à l'Imprimerie nationale*, 1990.

[5] Oren Patashnik. BibTeX*ing*, feb 1998.

[6] Leslie Lamport. *MakeIndex: An Index Processor For LaTeX*, February 1987.

[7] Yves Perrousseaux. *Manuel de Typographie française élémentaire.* Atelier Perrousseaux éditeur, 1995.

Indice analitico

Elenco delle tabelle

Elenco delle figure

Indice