

LaTeX



50.1	LaTeX: introduzione	468
50.1.1	Comandi e modelli sintattici	469
50.1.2	Opzioni e argomenti	470
50.1.3	Esempio iniziale	471
50.1.4	Sopravvivere nel caos	475
50.1.5	Struttura essenziale di un documento LaTeX 477	
50.1.6	Composizione per approssimazione successiva ...	481
50.2	Personalizzazione	483
50.2.1	Definizione e ridefinizione di macro	483
50.2.2	File di stile	488
50.3	Localizzazione	488
50.3.1	Terminologia	489
50.3.2	Sillabazione	489
50.3.3	Inconvenienti legati alla sillabazione	490
50.3.4	Spaziatura orizzontale	491
50.3.5	Codifica	491
50.3.6	Localizzazione in pratica	492
50.4	Variabili e tipi di dati	493
50.4.1	Lunghezze	493
50.4.2	Contatori	496
50.5	Struttura del documento	503

50.5.1	Dichiarazione dello stile generale	503
50.5.2	Preambolo	506
50.5.3	Inizio e fine del documento	506
50.5.4	Suddivisione del documento	508
50.5.5	Ambienti	515
50.6	LaTeX: la pagina	516
50.6.1	Dimensioni della pagina	516
50.6.2	Flusso verticale del testo, colonne e salti pagina	521
50.6.3	Stile della pagina	528
50.6.4	Numerazione delle pagine	530
50.6.5	Note a piè di pagina	531
50.6.6	Sillabazione	533
50.7	Caratteri	533
50.7.1	Caratteri che hanno un significato speciale	534
50.7.2	Corpi uniformi	536
50.7.3	Caratteri da stampa	538
50.7.4	Definizione dettagliata del carattere da stampa	543
50.7.5	Linee e cornicette attorno al testo	546
50.7.6	Spostamento verticale del testo	547
50.8	LaTeX: blocchi di testo	548
50.8.1	Caratteristiche normali di un paragrafo	548

50.8.2	Allineamento del testo	550
50.8.3	Elenchi	552
50.8.4	Citazioni	562
50.8.5	Versi	563
50.8.6	Testo letterale	565
50.8.7	Ambienti fluttuanti	567
50.9	Spazi e scatole	571
50.9.1	Spazi orizzontali	572
50.9.2	Spazi verticali	573
50.9.3	Elasticità controllabile	577
50.9.4	Interlinea	578
50.9.5	Scatole	580
50.9.6	Testo da salvare e da recuperare	587
50.9.7	Linee	588
50.10	Riferimenti incrociati	591
50.10.1	Indice generale, indice delle tabelle e delle figure	591
50.10.2	Riferimenti liberi	598
50.10.3	Indice analitico	601
50.10.4	Bibliografie	607
50.11	Tabelle	611
50.11.1	Tabella fatta di tabulazioni	611
50.11.2	Tabella fatta di colonne e righe	620
50.11.3	Tabella a cavallo di più pagine	630

50.12	LaTeX: figure	633
50.12.1	Disegnare con LaTeX	634
50.12.2	Importazione di file EPS esterni	643
50.12.3	Estrazione di file EPS dal file PostScript finale ...	645
50.13	Ambienti matematici	647
50.13.1	Definizione della modalità matematica	647
50.13.2	Spazi orizzontali e punteggiatura	650
50.13.3	Caratteri e simboli	651
50.13.4	Delimitatori	659
50.13.5	Modifica delle caratteristiche del carattere ..	661
50.13.6	Espressioni varie	663
50.13.7	Matrici e sistemi di equazioni	666
50.13.8	Teoremi, corollari e dichiarazioni simili	670
50.14	File esterni	676
50.14.1	Importazione	676
50.14.2	Importazione letterale	677
50.14.3	Creazione di file	677
50.15	Riferimenti	680

array 666 center 550 chapter 496 displaymath 647
document 477 enumerate 552 enumi 552 enumii 552
enumiii 552 enumiv 552 equation 496 647 figure 496
567 flushleft 550 flushright 550 footnote 496 531
itemize 552 list 552 longtable 630 lrbox 587
makeindex 601 math 647 page 496 530 paragraph 496

part 496 picture 634 quotation 562 quote 562 section
 496 subparagraph 496 subsection 496 subsubsection
 496 tabbing 611 table 496 567 567 tabular 620
 thebibliography 607 verbatim 565 verse 563 & 620 ^
 663 \ (647 \) 647 \= 611 \above 663 \addcontentsline
 591 \addtocontents 591 \addtocounter 496
 \addtolength 493 \addvspace 573 \Alph 496 \alph 496
 \appendix 477 508 \arabic 496 \atop 663 \author 477
 506 \baselineskip 548 \begin{document} 506
 \bfseries 538 \bibitem 607 \Big 659 \big 659 \Bigg
 659 \bigg 659 \Biggl 659 \biggl 659 \Biggr 659 \biggr
 659 \Bigl 659 \bigl 659 \Bigr 659 \bigr 659 \bigskip
 573 \bigskipamount 573 \caption 567 \centerline
 550 \chapter 477 508 \choose 663 \circle 634 \cite
 607 \cleardoublepage 521 567 \clearpage 521 567
 \cline 620 \closeout 677 \dashbox 634 \date 477 506
 \displaystyle 661 \documentclass 477 503 \dotfill
 572 \doublespacing 578 \emph 542 \endfoot 630
 \endhead 630 \end{document} 506 \epsfig 643
 \evensidemargin 516 \fbox 546 580 \fboxrule 580
 \fboxsep 580 \flushbottom 521 \fnsymbol 496
 \fontencoding 543 \fontfamily 543 \fontseries 543
 \fontshape 543 \fontsize 538 543 \footnote 531
 \footnotemark 531 \footnotesize 536
 \footnotetext 531 \footskip 516 \frac 663 \frame
 546 634 \framebox 580 \frenchspacing 491 492
 \headheight 516 \headsep 516 \hfill 572 \hline 620
 666 \hoffset 516 \hrulefill 572 \hspace 572 \Huge
 536 \huge 536 \hyphenation 533 \immediate 677

`\include` 676 `\includegraphics` 643 `\includeonly`
676 `\indent` 548 `\index` 601 `\input` 676 `\item` 552
`\itemsep` 552 `\itshape` 538 `\jobname` 677 `\kill` 611
`\label` 598 `\labelitemi` 552 `\labelitemii` 552
`\labelitemiii` 552 `\labelitemiv` 552 `\large` 536
`\LARGE` 536 `\Large` 536 `\leftskip` 548 `\line` 634
`\linebreak` 521 `\linespread` 578 `\linethickness` 634
`\linewidth` 548 `\listoffigures` 567 591
`\listoftables` 567 591 `\makebox` 580 634 `\makeindex`
601 `\maketitle` 477 506 `\markboth` 528 `\markright` 528
`\mathbf` 538 `\mathcal` 538 `\mathit` 538 `\mathrm` 538
`\mathsf` 538 `\mathtt` 538 `\mbox` 533 580 `\medskip` 573
`\medskipamount` 573 `\minipage` 580 `\newcommand` 483
`\newcounter` 496 `\newenvironment` 483 `\newlength`
493 `\newline` 521 `\newpage` 521 `\newtheorem` 670
`\newwrite` 677 `\noindent` 548 `\nopagebreak` 521
`\normalfont` 538 `\normalsize` 536 `\numberline` 591
`\oddsidemargin` 516 `\onecolumn` 521
`\onehalfspacing` 578 `\openout` 677 `\oval` 634 `\over`
663 `\overbrace` 659 `\overline` 546 659 `\pagebreak` 521
`\pagenumbering` 530 `\pageref` 598 `\pagestyle` 528
`\paperheight` 516 `\paperwidth` 516 `\par` 548
`\paragraph` 477 508 `\parbox` 580 611 `\parindent` 548
`\parskip` 548 `\part` 477 508 `\poptabs` 611 `\printindex`
601 `\pushtabs` 611 `\put` 634 `\raggedbottom` 521
`\raggedleft` 550 `\raggedright` 550 `\raisebox` 547
`\ref` 598 `\refstepcounter` 496 `\renewcommand` 483
`\renewenvironment` 483 `\rightskip` 548 `\rmfamily`
538 `\roman` 496 `\Roman` 496 `\rule` 588 `\samepage` 521

`\scriptscriptstyle` 661 `\scriptsize` 536
`\scriptstyle` 661 `\scshape` 538 `\section` 477 508
`\selectfont` 538 543 `\setcounter` 496 `\setlength` 493
`\settodepth` 493 `\settoheight` 493 `\settowidth` 493
`\sffamily` 538 `\singlespacing` 578 `\slshape` 538
`\small` 536 `\smallskip` 573 `\smallskipamount` 573
`\sqrt` 663 `\stackrel` 663 `\stepcounter` 496 `\stretch`
477 508 `\string` 677 `\subparagraph` 477 508 `\subsection`
477 508 `\subsubsection` 477 `\tableofcontents` 477 506
591 `\textasciicircum` 534 `\textasciitilde` 534
`\textbackslash` 534 `\textbf` 538 `\textheight` 516
`\textit` 538 `\textnormal` 538 `\textrm` 538 `\textsc` 538
538 `\textsf` 538 `\textsl` 538 `\textstyle` 661 `\texttt`
538 `\textup` 538 `\textwidth` 516 `\thanks` 506 `\the` 493
496 `\theenumi` 552 `\theenumii` 552 `\theenumiii` 552
`\theenumiv` 552 `\thispagestyle` 528 `\tiny` 536 `\title`
477 506 `\topmargin` 516 `\ttfamily` 538 `\twocolumn` 521
`\underbrace` 659 `\underline` 546 659 `\unitlength` 634
`\upshape` 538 `\usebox` 587 `\usefont` 543 `\usepackage`
477 `\usepackage{float}` 567 `\usepackage{latexsym}`
651 `\usepackage{longtable}` 630
`\usepackage{makeidx}` 601 `\usepackage{setspace}`
578 `\usepackage{verbatim}`
677 `\usepackage[italian]{babel}` 489
492 `\usepackage[T1]{fontenc}` 491 492
`\usepackage[utf8x]{inputenc}` 491 492 `\value` 496
`\vector` 634 `\verb` 565 `\verbatiminput` 677 `\vfill` 521
573 `\voffset` 516 `\vspace` 573 `\write` 677 `\#` 534 `\&` 534
`\` 611 620 `\{` 534 `\}` 534 `\$` 534 `_` 534 `\>` 611 `\[` 647 `\%` 534

\] 647 \$ 647 \$\$ 647 _ 663

50.1 LaTeX: introduzione

«

LaTeX è un pacchetto di macro per TeX molto comune, per facilitare la realizzazione di documenti tipici. Si fa riferimento implicito a questo pacchetto di macro utilizzando un comando specifico per la composizione, come già descritto nella sezione 49.1.3. Pertanto, di solito si ottiene la composizione di un sorgente TeX scritto usando macro LaTeX attraverso il comando seguente:

```
latex file_sorgente
```

Normalmente, il file sorgente scritto con macro LaTeX ha la solita estensione ‘.tex’; inoltre si usano le stesse procedure per la rielaborazione dei file DVI generati dalla composizione.

In generale, la scrittura di un sorgente LaTeX deve seguire le regole imposte dal suo pacchetto di macro. Ciò significa che le istruzioni o le macro del linguaggio TeX originale potrebbero creare dei conflitti, se l’uso di queste non è previsto espressamente.

Quando si usano solo le macro LaTeX, lasciando da parte TeX, queste macro possono essere viste come **comandi** del linguaggio LaTeX, mentre le **macro** diventano quelle definizioni create con il linguaggio LaTeX stesso.

50.1.1 Comandi e modelli sintattici

La necessità di avere il controllo della situazione, fa sì che LaTeX definisca anche un modo proprio per la creazione di altre macro; inoltre, esiste la possibilità di avere degli argomenti opzionali, che si indicano tra parentesi quadre. In generale, è difficile definire in modo completo come possono essere conformate le chiamate ai comandi e alle macro di LaTeX, comunque il modello sintattico seguente dovrebbe darne un'idea sufficiente per cominciare.

```
\comando [opzioni] ... {argomenti} ...
```

È già noto il fatto che le parentesi graffe fanno parte dell'istruzione; inoltre, le parentesi quadre devono essere usate per gli argomenti opzionali, ma se queste opzioni mancano, si omettono anche le parentesi quadre stesse. Per esempio, si può scrivere:

```
\documentclass[a4paper]{book}
```

oppure soltanto:

```
\documentclass{book}
```

Infine, c'è anche da considerare che alcuni comandi prevedono l'uso opzionale di un asterisco alla fine del loro nome, come nel caso della dichiarazione del titolo di una sezione:

```
\section{Bla bla bla}
```

Oppure:

```
\section*{Bla bla bla}
```

Il significato dell'asterisco dipende dal comando.

Ricapitolando, un modello sintattico riferito a comandi o macro di LaTeX deve essere letto in modo speciale: le parentesi graffe, le parentesi quadre e l'asterisco fanno parte dell'istruzione; inoltre, se non si indicano opzioni, le parentesi quadre che le delimitano non vanno usate.

50.1.2 Opzioni e argomenti

«

Fino a questo punto si è visto che le opzioni sono argomenti facoltativi che, se utilizzati, vanno delimitati attraverso delle parentesi quadre. La loro posizione è stabilita dalla sintassi del comando stesso, anche se di solito dovrebbero trovarsi prima degli argomenti normali.

Un comando potrebbe prevedere l'uso di più opzioni in sequenza, o alternate con gli argomenti. Ma oltre a questo, un'opzione potrebbe essere interpretata in modo da estrapolare più sotto-opzioni, delimitate generalmente attraverso una virgola.

Sugli argomenti c'è poco da aggiungere, tranne ripetere che il loro utilizzo è obbligatorio; inoltre, anche in questo caso, ci possono essere situazioni in cui un argomento è composto da più sotto-argomenti separati da virgole.

Segue la descrizione di alcuni esempi.

- `\documentclass{book}`

Definisce lo stile generale **'book'** per un documento che inizia.

- `\documentclass[a4paper]{book}`

Definisce lo stile generale **'book'**, specificando l'opzione **'a4paper'**.

- `\documentclass[a4paper,12pt]{book}`

Definisce lo stile generale **'book'**, specificando l'opzione **'a4paper, 12pt'**, che in pratica si traduce in due sotto-opzioni, distinte in base alla presenza della virgola.

- `\newcommand{\dattilografico}[1]{\texttt{#1}}`

Questo è un comando un po' difficile da interpretare (serve a creare una macro di LaTeX), comunque si può osservare che appare un'opzione dopo un argomento e prima dell'ultimo argomento. È il caso di precisare che in questo momento, **'\dattilografico'** e **'\texttt{#1}'** sono solo stringhe che hanno un qualche valore per il comando **'\newcommand'**.

- `\epsfig{file=prova,height=3cm,angle=0}`

Questo comando non riguarda direttamente LaTeX, ma proviene da un pacchetto che comunque lo accompagna. Come si può osservare, c'è solo un argomento, scomposto in tre sotto-argomenti separati da virgole.

50.1.3 Esempio iniziale

Prima di affrontare l'analisi della struttura del sorgente LaTeX, conviene vedere un esempio introduttivo abbastanza completo: «

```
\documentclass{article}

% Inizia il preambolo.

\setlength{\textwidth}{11cm}
\setlength{\textheight}{11cm}

% Fine del preambolo.
```

```
\begin{document}
```

```
% Inizia il documento vero e proprio.
```

```
\section{Introduzione a \LaTeX}
```

Questo \`e un esempio di documento scritto con \LaTeX .
Come si pu\`o vedere \`e gi\`a stato definito uno stile
generale del documento: article.

```
\subsection{Suddivisione del documento}
```

Lo stile article prevede una suddivisione in sezioni
sottosezioni ed eventuali sotto-sottosezioni.

```
\subsection{Paragrafi}
```

Il testo di un paragrafo termina quando nel sorgente viene
incontrata una riga vuota (una riga bianca).

Questo \`e l'inizio di un nuovo paragrafo e si nota perch\`e
la prima riga \`e leggermente rientrata.

```
\subsection{Gli ambienti}
```

\LaTeX utilizza gli ambienti per definire dei comportamenti
circondati a zone particolari del testo.

Per esempio, la centratura si ottiene utilizzando l'ambiente
center.

```
\begin{center}
```

Questo \`e un esempio di testo centrato.

```
\end{center}
```

```
% Fine del documento.
```

```
\end{document}
```

Supponendo di attribuire a questo file il nome ‘`primo.tex`’, si può procedere con la composizione nel modo seguente:

```
$ latex primo [Invio]
```

Se non vengono rilevati errori, durante l’elaborazione si vedono diverse informazioni sul procedimento della composizione, come già descritto nella sezione [49.1.3](#) a proposito di TeX. Nello stesso modo di TeX vengono generati dei file contenenti informazioni e la composizione nel file ‘`primo.dvi`’. Ma a differenza di TeX puro e semplice, si ottiene anche il file ‘`primo.aux`’, contenente delle porzioni di codice da riutilizzare nelle composizioni successive.

Il risultato della composizione di questo esempio si può vedere nella figura 50.11. La conversione del file DVI in qualcosa di più conveniente, è descritta nella sezione [26.3](#).

Figura 50.11. Il risultato della composizione del sorgente LaTeX di esempio.

1 Introduzione a L^AT_EX

Questo è un esempio di documento scritto con L^AT_EX. Come si può vedere è già stato definito uno stile generale del documento: `article`.

1.1 Suddivisione del documento

Lo stile `article` prevede una suddivisione in sezioni sottosezioni ed eventuali sotto-sottosezioni.

1.2 Paragrafi

Il testo di un paragrafo termina quando nel sorgente viene incontrata una riga vuota (una riga bianca).

Questo è l'inizio di un nuovo paragrafo e si nota perché la prima riga è leggermente rientrata.

1.3 Gli ambienti

L^AT_EX utilizza gli ambienti per definire dei comportamenti circoscritti a zone particolari del testo. Per esempio, la centratura si ottiene utilizzando l'ambiente `center`.

Questo è un esempio di testo centrato.

50.1.4 Sopravvivere nel caos

Nella sezione [49.1](#) è descritto in che modo si comporta TeX quando si presentano degli errori di composizione. Il pacchetto di macro che compone LaTeX aggiunge informazioni in fase di composizione. Per esempio, si può provare a realizzare un file contenente dei problemi, per vedere come si comporta LaTeX effettivamente:

```
\documentstyle{article}
\begin{document}
\section{Problemi con \LaTeX}
Quanti problemi con \LaTeX!
\end{document}
```

Leggendo il file delle registrazioni (il file con estensione ‘.log’) si può trovare un avvertimento che riguarda il comando ‘**\documentstyle**’, considerato superato e da non utilizzare.

```

      Entering LaTeX 2.09 COMPATIBILITY MODE
*****
!!WARNING!!      !!WARNING!!      !!WARNING!!      !!WARNING!!

This mode attempts to provide an emulation of the LaTeX 2.09
author environment so that OLD documents can be successfully
processed. It should NOT be used for NEW documents!

New documents should use Standard LaTeX conventions and start
with the \documentclass command.

Compatibility mode is UNLIKELY TO WORK with LaTeX 2.09 style
files that change any internal macros, especially not with
those that change the FONT SELECTION or OUTPUT ROUTINES.

Therefore such style files MUST BE UPDATED to use
      Current Standard LaTeX: LaTeX2e.

If you suspect that you may be using such a style file, which
```

```
is probably very, very old by now, then you should attempt to
get it updated by sending a copy of this error message to the
author of that file.
```

```
*****
```

Un avvertimento non è niente di eccezionalmente grave, soprattutto se poi non pregiudica la riuscita della composizione. Ma un avvertimento può segnalare il sorgere di un problema che più avanti può aggravarsi e diventare insuperabile. Se nell'esempio mostrato sopra si aggiunge un comando incompatibile si arriva al punto di crisi.

```
\documentstyle{article}
\usepackage{epsfig}
\begin{document}
\section{Problemi con \LaTeX}
Quanti problemi con \LaTeX!
\end{document}
```

Come si vede, è stato aggiunto il comando `'\usepackage{epsfig}'`, il cui scopo è solo quello di incorporare lo stile `'epsfig.sty'` che si trova da qualche parte, dove LaTeX può trovarlo.

Ciò che dovrebbe succedere è che lo stile richiesto sia incompatibile con una vecchia versione di LaTeX, oppure che sia incompatibile con il funzionamento che si impone a LaTeX quando si utilizza il comando `'\documentstyle'`.

```
! LaTeX Error: LaTeX2e command \usepackage in LaTeX 2.09 document.
```

```
See the LaTeX manual or LaTeX Companion for explanation.
```

```
Type H <return> for immediate help.
```

Ecco la segnalazione di errore. Fortunatamente, viene indicato precisamente il punto in cui appare l'errore, esattamente come succede

con la composizione in TeX puro e semplice: si tratta proprio del comando ‘`\usepackage`’ che appare nella riga numero due del testo sorgente.

```
1.2 \usepackage
      {epsfig}
```

Subito dopo appare l’invito consueto di TeX, che chiede all’utente di prendere una decisione al riguardo. Per il resto, tutto funziona come per TeX.

A differenza dell’utilizzo di TeX puro e semplice, può succedere che LaTeX mostri degli errori nell’ambito delle sue stesse macro, quando ci si spinge ai limiti delle possibilità di queste. Naturalmente, in questi casi occorre cercare il problema nelle istruzioni usate nel file sorgente, lasciando stare i file che compongono il pacchetto di macro di LaTeX.

50.1.5 Struttura essenziale di un documento LaTeX

Il sorgente di un documento scritto in TeX, utilizzando il pacchetto di macro LaTeX, ha una struttura che segue delle regole precise. La prima cosa a essere definita è il tipo di documento, ovvero lo stile generale a cui si vuole fare riferimento. A questo segue eventualmente un preambolo, cioè l’indicazione più o meno facoltativa di altri elementi stilistici insieme alle informazioni che servono a comporre il titolo del documento. Quindi inizia il documento vero e proprio.

Lo stile generale del documento viene definito all’inizio del sorgente LaTeX attraverso la dichiarazione seguente:

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

Per esempio:

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

definisce l'uso della classe '**book**', utilizzando un foglio A4 con il corpo normale dei caratteri a 11 punti tipografici; mentre il comando

```
\documentclass{book}
```

definisce l'uso della classe '**book**', senza opzioni, che così viene impostata in modo predefinito.

Il preambolo è quella parte di sorgente LaTeX che sta tra la dichiarazione della classe (o dello stile generale) e la dichiarazione di inizio del documento. Normalmente viene usata per specificare l'utilizzo di stili aggiuntivi e per l'inserimento di tutti quegli elementi che compongono il titolo del documento e gli indici eventuali.

Una dichiarazione molto importante del preambolo è l'inclusione di uno stile aggiuntivo, secondo la sintassi seguente:

```
\usepackage[opzioni]{pacchetto}
```

Le opzioni utilizzabili dipendono dal tipo particolare di stile a cui si fa riferimento. Un file di stile può anche essere scritto dall'utilizzatore, solitamente partendo da un altro già esistente.

```
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
```

L'esempio mostra l'inclusione del pacchetto '**inputenc**' allo scopo di ammettere la codifica dei caratteri UTF-8 nel sorgente LaTeX, as-

sieme al pacchetto ‘**fontenc**’ per ottenere una composizione con un tipo di carattere che contenga le lettere accentate e i simboli speciali più importanti utilizzati in Europa.

Così come è possibile aggiungere altri stili, è possibile utilizzare direttamente delle dichiarazioni riferite a singoli elementi stilistici. Nell’esempio presentato all’inizio si utilizzavano due dichiarazioni:

```
\setlength{\textwidth}{11cm}  
\setlength{\textheight}{11cm}
```

In questo caso, si definiva la larghezza e l’altezza del testo, senza fare riferimento a un formato standard.

Il preambolo serve anche per definire gli elementi che fanno parte del titolo del documento. Questi dipendono dal tipo di stile generale utilizzato, ma di solito comprendono almeno il titolo, l’autore e la data, come nell’esempio seguente:

```
\title{Usare \LaTeX}  
\author{Pinco Pallino}  
\date{11/11/2011}
```

L’inizio del documento è contrassegnato dalla dichiarazione ‘**\begin{document}**’ e la fine da ‘**\end{document}**’. Tutto quello che appare dopo la conclusione del documento viene semplicemente ignorato.

Subito dopo l’apertura del documento viene collocata normalmente l’istruzione di creazione del titolo, ‘**\maketitle**’, seguita eventualmente da quello di creazione dell’indice, ‘**\tableofcontents**’.

```
\begin{document}  
\maketitle  
\tableofcontents  
...  
...  
\end{document}
```

Il corpo del documento può essere normalmente suddiviso, a seconda del tipo di classe utilizzato.

```
\livello_di_suddivisione [opzioni_eventuali] {titolo_della_suddivisione}
```

I nomi dei livelli di suddivisione possono essere i seguenti, elencati in ordine decrescente di importanza:

- **‘part’**
- **‘chapter’**
(solo per le classi **‘book’** e **‘report’**)
- **‘section’**
- **‘subsection’**
- **‘subsubsection’**
- **‘paragraph’**
- **‘subparagraph’**

Per esempio,

```
\section{Introduzione a \LaTeX}
```

definisce l’inizio di una sezione che ha il titolo indicato tra le parentesi graffe.¹

In particolare esiste un comando speciale, ‘`\appendix`’ che viene utilizzato così, senza opzioni, esclusivamente per modificare il modo in cui vengono numerate le suddivisioni, che da quel punto vengono trattate come parte di un’appendice.

```
\chapter{Bla bla bla}
...
\appendix
\chapter{Appendice bla bla}
...
```

L’esempio mostra proprio questo: il capitolo denominato «Appendice bla bla» è la prima appendice.

50.1.6 Composizione per approssimazione successiva

Tra i vari automatismi offerti da LaTeX, esiste la possibilità di generare degli indici e dei riferimenti incrociati. Per ottenere questo risultato, LaTeX deve annotare inizialmente la posizione dei riferimenti da leggere poi in una fase successiva. «

Queste annotazioni vengono fatte in file ausiliari, con estensione ‘.aux’ e altre, che vengono letti automaticamente attraverso la ripetizione del processo di composizione. Osservando il file ausiliario principale generato dalla composizione dell’esempio iniziale, si può notare che si tratta di istruzioni LaTeX:

```

\relax
\@writefile{toc}{\contentsline {section}{\numberline ←
↪{1}Introduzione a \LaTeX  }{1}}
\@writefile{toc}{\contentsline {subsection}{\numberline ←
↪{1.1}Suddivisione del documento}{1}}
\@writefile{toc}{\contentsline {subsection}{\numberline ←
↪{1.2}Paragrafi}{1}}
\@writefile{toc}{\contentsline {subsection}{\numberline ←
↪{1.3}Gli ambienti}{1}}

```

Nel caso particolare dell'esempio mostrato, si tratta solo di istruzioni necessarie alla costruzione dell'indice generale; intuitivamente si comprende il significato dei dati che appaiono: il numero della sezione, il titolo e la pagina in cui si trova.

Gli indici e i riferimenti incrociati vanno a modificare l'impaginazione; si pensi al caso dell'indice generale, dapprima assente del tutto, che in una seconda fase può impegnare da solo più pagine. Pertanto, sono indispensabili tre rielaborazioni per arrivare a una composizione ragionevolmente corretta: la prima fase genera i dati che inizialmente sono assenti del tutto; la seconda incorpora i dati, che però si riferiscono all'impaginazione precedente, in cui mancavano, rigenerando una nuova serie di dati aggiornati; la terza rielaborazione utilizza i dati aggiornati nella seconda fase. Teoricamente, una quarta elaborazione potrebbe generare un risultato ancora più sicuro, che però di solito non risulta necessario.

È importante chiarire che LaTeX non dispone di un procedimento automatico in grado di avviare e riavviare la composizione per tutte le volte in cui ciò possa risultare necessario, pertanto è l'utente stesso che riavvia la composizione in base alle esigenze:

```
$ latex file_sorgente [Invio]
```

```
$ latex file_sorgente [Invio]
```

```
$ latex file_sorgente [Invio]
```

50.2 Personalizzazione

Come accennato all'inizio del capitolo, LaTeX mette a disposizione dei comandi appositi per la creazione di nuove macro. In queste sezioni viene mostrato come creare e ridefinire una macro di LaTeX e come importare un file di stile, ovvero un file contenente delle macro.

In particolare, negli esempi si fa riferimento spesso a comandi comuni di LaTeX che però non sono ancora stati descritti. In questa fase è sufficiente comprendere i meccanismi con cui si definiscono le macro e si utilizzano i file esterni.

50.2.1 Definizione e ridefinizione di macro

Nel momento in cui si lavora con documenti di grandi dimensioni, oppure si sta preparando una *veste grafica* per le proprie pubblicazioni, è importante creare una serie di istruzioni personalizzate per la creazione di ambienti, anche se queste non sono altro che una copia di istruzioni già esistenti. Il vantaggio di questo modo di procedere sta nella possibilità successiva di cambiare tutta la veste

grafica semplicemente modificando il funzionamento delle istruzioni personalizzate.

LaTeX prevede anche la possibilità di ridefinire istruzioni già esistenti, ma in tal caso è importante attribuire un senso particolare (e personale) a quelle istruzioni.

L'istruzione '`\newcommand`' permette di creare un comando nuovo, mentre '`\renewcommand`' permette di ridefinirne uno già dichiarato in precedenza.

```
\newcommand{comando} [n_argomenti] {definizione}
```

```
\newcommand{comando} [n_argomenti] [primo_argomento_fisso] ↔  
↔ {definizione}
```

```
\renewcommand{comando} [n_argomenti] {definizione}
```

```
\renewcommand{comando} [n_argomenti] [primo_argomento_fisso] ↔  
↔ {definizione}
```

Se il comando può avere degli argomenti, è necessario indicarne la quantità, attraverso una sola cifra numerica da uno a nove; inoltre, è possibile specificare un contenuto fisso del primo argomento, che viene aggiunto al primo argomento della chiamata. Quando si prevedono degli argomenti, questi vengono inseriti nella stringa di definizione del comando utilizzando delle metavariables nella forma '`#n`', dove il numero rappresenta l'*n*-esimo argomento.

In generale, non c'è alcun motivo di usare la dichiarazione in cui si prevede un primo argomento fisso, dal momento che quel testo può essere inserito direttamente nella definizione del comando.

Una volta dichiarato o ridichiarato il comando, questo può essere utilizzato attraverso una sintassi riassumibile nel modo seguente, dove gli argomenti eventuali, appaiono tra parentesi graffe.

```
\comando | \comando {argomento}...
```

L'utilizzo più semplice, riguarda la definizione di un comando senza argomenti, come nell'esempio seguente, dove viene dichiarato '`\bftt`' il cui scopo è quello di iniziare l'uso di un carattere neretto e dattilografico.

```
\newcommand{\bftt}{\bfseries\ttfamily}
```

Come si vede, lo scopo è solo quello di sostituire a '`\bftt`', i comandi '`\bfseries`' e '`\ttfamily`'. Quello che segue è un esempio di come potrebbe essere utilizzato.

```
Il file \bftt mio-file \normalfont contiene...
```

Secondo l'esempio, la parola «mio-file» viene evidenziata in neretto e dattilografico, mentre subito dopo, attraverso il comando '`\normalfont`', riprende lo stile normale.

L'esempio seguente definisce un comando che richiede un argomento. Ciò che si ottiene serve a delimitare una zona in neretto dattilografico.

```
\newcommand{\bftt}[1]{\textbf{\texttt{#1}}}
```

Lo scopo è quello di sostituire al comando ‘`\bftt{...}`’, i comandi ‘`\textbf{\texttt{...}}`’. Quello che segue è un esempio di come potrebbe essere utilizzato; si può osservare che in questo caso non occorre riconvertire il testo dopo la zona delimitata con le parentesi graffe.

```
Il file \bftt{mio-file} contiene...
```

L’esempio seguente, utilizzando sempre un solo argomento, ha lo scopo di replicarlo cambiandone leggermente lo stile.

```
\newcommand{\triplo}[1]{\textit{#1} \textbf{#1} \texttt{#1}}
```

Utilizzandolo nel modo seguente, si ottiene la ripetizione della parola «tanto» per tre volte, ognuna con uno stile differente: corsivo, neretto e dattilografico.

```
Ti amo \triplo{tanto}...
```

Prima di proseguire, vale la pena di vedere un esempio in cui si dichiara un comando che prevede l’uso di più argomenti.

```
\newcommand{\somma}[3]{\texttt{#1}+\texttt{#2}=\texttt{#3}}
```

Utilizzando il comando appena creato nel modo seguente, si ottiene esattamente il testo ‘**5+6=11**’, dove i numeri sono in dattilografico e i segni sono composti con caratteri normali.

```
\somma{5}{6}{11}
```

Oltre a questo è possibile definire, o ridefinire, degli ambienti da utilizzare nei comandi ‘`\begin{...}`’ e ‘`\end{...}`’. Per queste operazioni si utilizzano le istruzioni ‘`\newenvironment`’ e ‘`\renewenvironment`’.

```
\newenvironment{ambiente}[n_argomenti]{def_iniziale}{def_finale}
```

```
\newenvironment{ambiente}[n_argomenti][argomento_predefinito]{  
↵  
↵}{def_iniziale}{def_finale}
```

```
\renewenvironment{ambiente}[n_argomenti]{def_iniziale}{def_finale}
```

Se la dichiarazione del nuovo ambiente può avere degli argomenti, è necessario indicarne il numero, attraverso una sola cifra numerica da uno a nove. Questi possono essere utilizzati **solo** nella definizione di inizio, attraverso le metavariabili ‘*#n*’. Come nel caso della dichiarazione di un nuovo comando, è possibile specificare il primo argomento predefinito, da utilizzare nel caso non ne sia fornito alcuno.

Una volta dichiarato o ridefinito l’ambiente, questo può essere utilizzato attraverso una sintassi riassumibile nel modo seguente, dove gli argomenti eventuali, appaiono tra parentesi graffe.

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

```
\begin{ambiente}{argomento}...
testo_contenuto
...
\end{ambiente}
```

L'utilizzo più semplice riguarda la definizione di un ambiente senza argomenti, come nell'esempio seguente, dove viene dichiarato l'ambiente '`\bftt`' il cui scopo è quello di iniziare l'uso di un carattere neretto e dattilografico.

```
\newenvironment{bftt}{\bfseries\ttfamily}{\normalfont}
```

Come si vede, si intende sostituire '`\begin{bftt}`' con il comando '`\bfseries\ttfamily`' e rimpiazzare '`\end{bftt}`' con il comando '`\normalfont`'.

50.2.2 File di stile



La personalizzazione di istruzioni LaTeX può avvenire all'interno del documento stesso, ma generalmente è preferibile creare un file di stile da includere con l'istruzione '`\usepackage{file_di_stile}`'.

Quando si crea un nuovo stile conviene fare una copia di uno di quelli già utilizzati da LaTeX e quindi modificarlo.

Quando si utilizza TeTeX, questi file di stile dovrebbero trovarsi nella directory '`texmf/tex/latex/base/`'.

50.3 Localizzazione



I problemi legati alla localizzazione del funzionamento di LaTeX riguardano in particolare i termini generati automaticamente dalla

composizione (come *Chapter*, *Index* e simili) e la separazione in sillabe.

50.3.1 Terminologia

A seconda dello stile generale del documento che si scrive, quasi sempre, il risultato finale contiene parole inserite automaticamente da LaTeX. Questi termini sono definiti all'interno del file di stile che identifica la classe del documento.

Per modificare questo comportamento si può utilizzare uno stile aggiuntivo, scelto tra quelli contenuti nella directory `'texmf/tex/generic/babel/'`, oppure si può creare uno stile personalizzato in cui si ridefiniscono le istruzioni che dichiarano questi termini.

Lo stile aggiuntivo viene caricato normalmente con il comando seguente, posto nel preambolo del documento.

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

50.3.2 Sillabazione

La sillabazione è configurata attraverso il file `'texmf/tex/generic/config/language.dat'`, il quale a sua volta fa riferimento a file contenuti in `'texmf/tex/generic/hyphen/'`.

Se all'interno di questo file sembra non essere attivata la sillabazione per la lingua italiana, conviene modificarlo attraverso il sistema di configurazione di TeX, descritto nella sezione [49.1](#).

Se la lingua italiana, o quella che interessa, risulta già attivata, oppure se è stata fatta la procedura per attivarla, si può controllare nel file `'texmf/web2c/latex.log'`, soprattutto per determinare il numero corrispondente che gli è stato assegnato. Se si trova una riga

simile a quella seguente, significa che la sillabazione in italiano è disponibile.

```
\l@italian=\language3
```

50.3.3 Inconvenienti legati alla sillabazione

«

Se quello che si scrive è un documento tecnico pieno di termini che non fanno parte della lingua italiana, forse conviene disabilitare la sillabazione per evitare la suddivisione di termini stranieri in modo errato. Per farlo, dovrebbe essere sufficiente prevedere l'uso della sillabazione nulla (*nohyphenation*), attivandola nel documento esattamente come si farebbe per un linguaggio normale.

Per esempio, se nel file ‘`texmf/web2c/latex.log`’ si trova la riga seguente,

```
\l@nohyphenation=\language4
```

molto probabilmente dovrebbe essere sufficiente utilizzare il comando ‘`\language4`’ nel documento. Se però, per qualche ragione questo non dovesse funzionare, si possono sempre usare metodi drastici: configurare il file ‘`texmf/tex/generic/config/language.dat`’ commentando tutte le direttive e annullando in ogni caso il sistema di sillabazione.

Eventualmente, se il problema è solo quello di evitare che qualche termine sia separato in sillabe, lo si può inserire in una scatola (come viene descritto meglio in seguito):

```
\mbox{parola}
```

50.3.4 Spaziatura orizzontale

Secondo una regola della tipografia del passato, ormai condannata generalmente, è necessario aumentare lo spazio che divide la fine di un periodo dall'inizio del successivo. Per qualche ragione si trovano ancora documenti in lingua inglese che seguono questa regola, anche quando si tratta di file di testo.

Purtroppo TeX segue quella filosofia e tende a rendere più grande lo spazio orizzontale che c'è tra un punto finale e la parola successiva se questa ha l'iniziale maiuscola. Oltre a tutto, questo sistema crea delle difficoltà nella scrittura degli acronimi o delle abbreviazioni. Si pone rimedio utilizzando anche con LaTeX il comando `'\frenchspacing'` nel preambolo del documento (si veda anche la sezione [49.12](#) a questo proposito).

50.3.5 Codifica

LaTeX permette l'uso di diverse codifiche, cioè diverse sequenze di simboli nei tipi di carattere utilizzati. Per la precisione, si distingue tra la codifica usata nel sorgente e il tipo di insieme di caratteri da usare. Nel primo caso, si usa normalmente il pacchetto `'inputenc'`, nel modo seguente, all'interno del preambolo, in modo da stabilire l'uso della codifica UTF-8 nel sorgente:

```
\usepackage[utf8x]{inputenc}
```

Per quanto riguarda l'insieme di caratteri, il tipo più vecchio è OT1, definito anche *TeX text*; il più recente e più usato è T1, definito anche *TeX text extended*. L'utilizzo dell'insieme di caratteri T1 è necessario se si vuole scrivere un documento che nel sorgente fa uso delle lettere accentate delle lingue europee occidentali.

La selezione dell'insieme di caratteri TeX avviene attraverso il caricamento del pacchetto '**fontenc**', indicando come opzione la sigla dell'insieme desiderato.

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

L'esempio mostra il caricamento dell'insieme di caratteri T1, che è quello che dovrebbe essere utilizzato nella maggior parte dei casi.

50.3.6 Localizzazione in pratica

«

La definizione del sistema di sillabazione è sempre necessario, mentre si è accennato poco sopra al problema dei termini da tradurre. Il modo più semplice per risolvere il problema della localizzazione (dopo avere sistemato la sillabazione) è quello di utilizzare le istruzioni seguenti nel preambolo.

```
\documentclass...  
...  
\usepackage[italian]{babel}  
\usepackage[utf8x]{inputenc}  
\usepackage[T1]{fontenc}  
\frenchspacing  
...  
\begin{document}  
...
```

In questo modo, se è stata definita una sillabazione italiana, questa viene attivata automaticamente; i termini come «capitolo», «pagina»,... sono tradotti in italiano; la codifica del sorgente è UTF-8, avendo richiesto l'insieme di caratteri tipografici T1 (quindi il sorgente può essere scritto utilizzando le lettere accentate senza la necessità di utilizzare codici macro particolari).

Se per qualche ragione si vuole redigere un testo multilingua, è possibile utilizzare il pacchetto ‘**babel**’ con l’indicazione di più linguaggi, come nel modo seguente:

```
\usepackage[italian,english]{babel}
```

Successivamente, per selezionarne uno, basta usare il comando ‘**\selectlanguage**’, con il nome prescelto. Ciò ha effetto fino all’uso di un altro comando ‘**\selectlanguage**’ che cambi tale indicazione.

```
...
\selectlanguage{italian}
...
\selectlanguage{english}
...
```

È importante non confondere ‘**\selectlanguage**’ con ‘**\language**’. Nel primo caso si fa riferimento a un comando del pacchetto ‘**babel**’; nel secondo si fa riferimento alla sillabazione. Il primo è in grado di condizionare il secondo, ma non viceversa: non esiste un «linguaggio» corrispondente alla sillabazione nulla. Se si intende definire la sillabazione nulla occorre passare per un comando ‘**\language**’ conforme.

50.4 Variabili e tipi di dati

Nella sezione [49.3](#) è descritto l’uso delle variabili nel linguaggio TeX, ma quando si usa LaTeX è bene utilizzare comandi appositi, ignorando la gestione che c’è a basso livello. A ogni modo, rimane la possibilità di usare l’istruzione ‘**\the**’ per trasformare il contenuto di una variabile LaTeX in testo della composizione finale.

50.4.1 Lunghezze



Le variabili adibite a contenere le lunghezze, si creano secondo LaTeX attraverso il comando ‘`\newlength`’:

```
\newlength\nome_lunghezza
```

Successivamente, si assegna un valore alle variabili di questo tipo con il comando ‘`\setlength`’:

```
\setlength{\nome_lunghezza}{lunghezza}
```

Come per TeX, le lunghezze che si assegnano in questo modo a una variabile, possono essere rigide oppure elastiche, secondo la stessa notazione di TeX:

```
lunghezza_richiesta [plus lunghezza_in_estensione] ←  
↔ [minus lunghezza_in_contraZIONE]
```

La lunghezza che costituisce il secondo argomento del comando ‘`\setlength`’, può essere il comando corrispondente a una lunghezza già definita in precedenza; in questo modo si può salvare un valore da richiamare in un momento successivo, quando si intende variare temporaneamente la dimensione di una lunghezza già esistente. Per esempio:

```
\newlength\salvataggio  
\setlength{\salvataggio}{\textwidth}  
\setlength{\textwidth}{10cm}  
...  
\setlength{\textwidth}{\salvataggio}
```

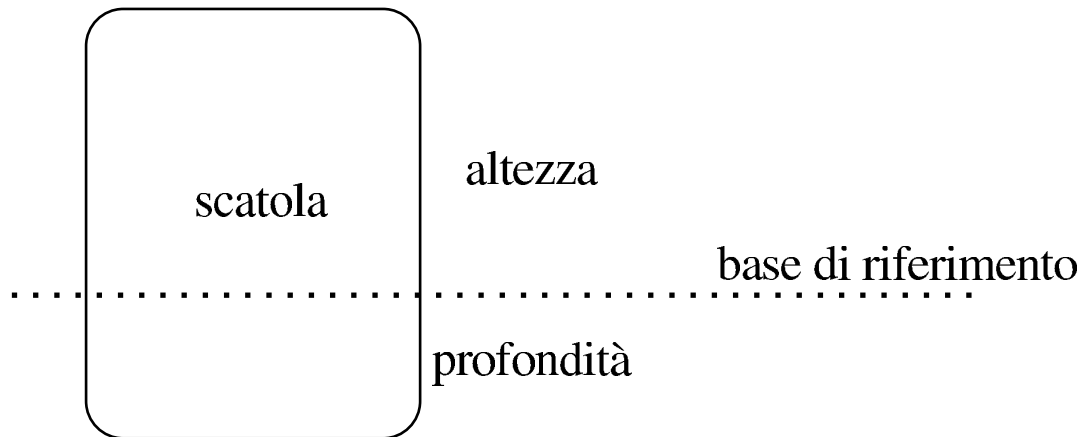
Quanto appare sopra, mostra il salvataggio della lunghezza originale di ‘`\textwidth`’ nella variabile ‘`\salvataggio`’, quindi la modifica del valore di ‘`\textwidth`’ temporaneamente, alla fine il ripristino del valore di ‘`\textwidth`’ leggendo il contenuto di ‘`\salvataggio`’.

L’elenco seguente riepiloga i comandi utili per la creazione e la modifica delle lunghezze:

<code>\newlength\nome_lunghezza</code>	crea una lunghezza nuova, corrispondente al comando ‘ <code>\nome_lunghezza</code> ’;
<code>\setlength{\nome_lunghezza}{lunghezza}</code>	assegna alla lunghezza indicata il valore dell’ultimo argomento, che può essere sia una lunghezza rigida, sia una lunghezza elastica;
<code>\addtolength{\nome_lunghezza} ← ↔{lunghezza_rigida}</code>	aggiunge alla lunghezza indicata il valore dell’ultimo argomento, che deve essere una lunghezza rigida;
<code>\settodepth{\nome_lunghezza}{testo}</code>	assegna alla lunghezza il valore della profondità del testo che appare come ultimo argomento;
<code>\settoheight{\nome_lunghezza}{testo}</code>	assegna alla lunghezza il valore dell’altezza del testo che appare come ultimo argomento;

<pre>\settowidth{\nome_lunghezza}{testo}</pre>	assegna alla lunghezza il valore della larghezza del testo che appare come ultimo argomento.
------------------------------------------------	----------------------------------------------------------------------------------------------

Figura 50.45. Dimensioni di una scatola.



50.4.2 Contatori

«

LaTeX gestisce dei contatori a cui si accede attraverso parole chiave, nell'ambito di comandi che ne consentono la creazione, la modifica e la lettura. L'elenco seguente riguarda una parte dei comandi coinvolti nella gestione dei contatori:

Comando	Descrizione
<pre>\newcounter{nome_contatore} ← ↔ [contatore_già_esistente]</pre>	crea un contatore con il nome indicato, il cui valore viene azzerato ogni volta che quello opzionale (tra parentesi quadre) viene incrementato;

Comando	Descrizione
<code>\setcounter{<i>nome_contatore</i>}{<i>valore</i>}</code>	assegna il valore indicato al contatore (deve trattarsi di un valore intero, che eventualmente può essere negativo);
<code>\stepcounter{<i>nome_contatore</i>}</code>	incrementa il contatore di una singola unità e azzera eventualmente i contatori che dipendono da questo;
<code>\refstepcounter{<i>nome_contatore</i>}</code>	si comporta come ‘ <code>\stepcounter</code> ’, con la differenza che coinvolge la creazione di un riferimento se seguito dal comando ‘ <code>\label</code> ’;
<code>\addtocounter{<i>nome_contatore</i>}{<i>valore</i>}</code>	aggiunge al contatore il valore indicato (deve trattarsi di un valore intero, che eventualmente può essere negativo);
<code>\arabic{<i>nome_contatore</i>}</code>	traduce il valore del contatore in un numero arabo nella composizione finale;

Comando	Descrizione
\the <i>nome_contatore</i>	quando viene creato un contatore, si crea implicitamente questo comando, con il quale si ottiene il valore del contatore nella composizione finale, espresso in modo predefinito (di solito si tratta di un numero arabo);
\alph{ <i>nome_contatore</i> }	traduce il valore del contatore in una lettera minuscola singola, pertanto si possono rappresentare solo valori da 1 a 26;
\Alph{ <i>nome_contatore</i> }	traduce il valore del contatore in una lettera maiuscola singola, pertanto si possono rappresentare solo valori da 1 a 26;
\roman{ <i>nome_contatore</i> }	traduce il valore del contatore in un numero romano con lettere minuscole, pertanto non si possono rappresentare valori negativi;

Comando	Descrizione
<code>\Roman{<i>nome_contatore</i>}</code>	traduce il valore del contatore in un numero romano con lettere maiuscole, pertanto non si possono rappresentare valori negativi;
<code>\fnsymbol{<i>nome_contatore</i>}</code>	traduce il valore del contatore in un simbolo, ma sono disponibili solo nove simboli, pertanto si rappresentano valori da 1 a 9;
<code>\value{<i>nome_contatore</i>}</code>	ottiene il valore del contatore, da usare all'interno di un'espressione (non riguarda la composizione).

Per esempio, si crea un contatore ‘**prova**’ con il comando seguente, assegnandogli implicitamente il valore iniziale zero:

```
\newcounter{prova}
```

Al contatore è possibile assegnare esplicitamente un valore:

```
\setcounter{prova}{-35}
```

In alternativa, si può assegnare al contatore il valore di un contatore già esistente; in questo caso si assegna il valore del contatore della pagina:

```
\setcounter{prova}{\value{page}}
```

Infine, si può inserire il valore del contatore nella composizione finale, per esempio in forma di numero arabo normale:

```
\arabic{prova}
```

Per comprendere la dipendenza che ci può essere tra i contatori, viene proposto un esempio più articolato, seguito dal risultato che si ottiene con la composizione:

```
\newcounter{uno}
\newcounter{due}[uno]

\stepcounter{uno}

\theuno . \thedue

\stepcounter{due}

\theuno . \thedue

\stepcounter{due}

\theuno . \thedue

\stepcounter{uno}

\theuno . \thedue
```

Come si può notare, in questo caso è stata usata la forma ‘**\thecontatore**’ per visualizzare il valore del contatore preso in esame:

```
1.0
1.1
1.2
2.0
```

Per quanto riguarda l’uso di ‘**\refstepcounter**’, le cose si complicano e richiedono la conoscenza dei comandi ‘**\label**’ e ‘**\ref**’

(sezione [50.10](#)). Qui viene mostrato un esempio che dovrebbe consentire una comprensione intuitiva della cosa:

```
\newcounter{prova}
\stepcounter{prova}
Il contatore ha raggiunto il valore \arabic{prova}.\par
\stepcounter{prova}
Il contatore ha raggiunto il valore \arabic{prova}.\par
\refstepcounter{prova}\label{qui}
Ciao.\par
\stepcounter{prova}
Il contatore ha raggiunto il valore \arabic{prova}.\par

Quando si mostrava la parola ``ciao'', il contatore aveva il
valore \ref{qui}.
```

Si osservi che in questo caso, avendo a che fare con riferimenti incrociati, la composizione deve essere eseguita almeno due volte:

Il contatore ha raggiunto il valore 1.

Il contatore ha raggiunto il valore 2.

Ciao.

Il contatore ha raggiunto il valore 4.

Quando si mostrava la parola “ciao”, il contatore aveva il valore 3.

Il comando ‘`\thecontatore`’ può essere ridefinito, se si desidera ottenere una forma di visualizzazione standard differente:

```
\newcounter{prova}
\renewcommand{\theprova}{\Roman{prova}}
\stepcounter{prova}
Il contatore ha raggiunto il valore \theprova.\par
\stepcounter{prova}
Il contatore ha raggiunto il valore \theprova.\par
```

Il contatore ha raggiunto il valore I.

Il contatore ha raggiunto il valore II.

LaTeX definisce e gestisce automaticamente un certo numero di contatori, a cui può essere conveniente fare riferimento. La tabella seguente elenca i contatori predefiniti più importanti:

Contatore	Descrizione
<code>part</code>	contatore delle parti;
<code>chapter</code>	contatore dei capitoli;
<code>section</code>	contatore delle sezioni;
<code>subsection</code>	contatore delle sottosezioni;
<code>subsubsection</code>	contatore delle sotto-sottosezioni;
<code>paragraph</code>	contatore dei paragrafi;
<code>subparagraph</code>	contatore dei sottoparagrafi;
<code>page</code>	contatore delle pagine;
<code>equation</code>	contatore delle equazioni;
<code>figure</code>	contatore delle figure;
<code>table</code>	contatore delle tabelle;
<code>footnote</code>	contatore delle note a piè di pagina;

Contatore	Descrizione
enumi	contatori usati negli elenchi numerati.
enumii	
enumiii	
enumiv	

50.5 Struttura del documento

LaTeX semplifica la scrittura di un documento, rispetto al linguaggio TeX puro e semplice, ma per questo richiede l'uso di una struttura particolare nel sorgente.

50.5.1 Dichiarazione dello stile generale

Lo stile generale del documento viene definito all'inizio del sorgente LaTeX attraverso la dichiarazione seguente:

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

Le classi possono essere:

Classe	Descrizione
slides	rappresenta una sequenza di diapositive (presentazioni).
letter	rappresenta una lettera vera e propria, dove non si ammette la separazione del testo in sezioni;
article	corrispondente a un documento senza tante pretese, ma ugualmente strutturato, che può ammettere la suddivisione in parti e sezioni (sono esclusi i capitoli);

Classe	Descrizione
report	corrispondente a un documento strutturato in parti e capitoli, inteso come una relazione;
book	permette di realizzare un libro suddiviso in parti e capitoli, ma a differenza della classe ' report ', le parti e i capitoli iniziano su pagine dispari (destre);

Il comando '**\documentclass**' ammette l'uso di una sola opzione, ma al suo interno possono essere indicate diverse sotto-opzioni, rappresentate da delle parole chiave, separate attraverso una virgola. Ogni classe di documento può gestire il suo gruppo particolare di sotto-opzioni, ma in generale, sono disponibili quelle seguenti che dovrebbero essere valide in ogni circostanza.

Dimensione dei caratteri:

Opzione	Descrizione
10pt	il testo normale usa caratteri con un corpo di 10 punti (predefinito);
11pt	il testo normale usa caratteri con un corpo di 11 punti;
12pt	il testo normale usa caratteri con un corpo di 12 punti.

Dimensione del foglio:

Opzione	Descrizione
a4paper	formato A4;
a5paper	formato A5;
b5paper	formato B5;

Opzione	Descrizione
letterpaper	formato Lettera (8,5 in × 11 in);
legalpaper	formato Legal (8,5 in × 14 in);
executivepaper	formato Executive (7,25 in × 10,5 in);.

Altre caratteristiche:

Opzione	Descrizione
portrait	orientamento verticale della pagina (predefinito);
landscape	orientamento orizzontale della pagina;
draft	bozza, evidenzia con una riga verticale il testo che non rispetta i bordi;
final	versione finale, senza segnalazioni sul debordamento del testo (predefinito);
twocolumn	divide il corpo in due colonne.

Per esempio:

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

definisce l'uso della classe **'book'**, utilizzando un foglio A4 con il corpo normale dei caratteri a 11 punti tipografici; mentre il comando

```
\documentclass{book}
```

definisce l'uso della classe **'book'**, senza opzioni, che così viene impostata in modo predefinito.

50.5.2 Preambolo

«

Il preambolo è quella parte di sorgente LaTeX che sta tra la dichiarazione della classe (o dello stile generale) e la dichiarazione di inizio del documento. In questo spazio si possono inserire dei comandi di LaTeX, ma non del testo da comporre. In particolare si inseriscono qui le dichiarazioni di stili aggiuntivi da incorporare, con il comando ‘`\usepackage`’:

```
\usepackage[opzioni]{pacchetto}
```

Un gruppo abbastanza comune di comandi che di norma vengono inseriti nel preambolo, è quello con cui si definiscono i dati necessari a costruire poi la pagina del titolo:

Comando	Descrizione
<code>\title{<i>titolo</i>}</code>	titolo del documento (si può usare la sequenza ‘ <code>\</code> ’ per ottenere un’interruzione di riga);
<code>\author{<i>autore</i>}</code>	autore o autori del documento (si può usare la sequenza ‘ <code>\</code> ’ per ottenere un’interruzione di riga);
<code>\date{<i>data</i>}</code>	data del documento;
<code>\thanks{<i>annotazione</i>}</code>	nota a piè di pagina riferita al titolo.

50.5.3 Inizio e fine del documento

«

L’inizio del documento è contrassegnato dalla dichiarazione ‘`\begin{document}`’ e la fine da ‘`\end{document}`’. Tutto

quello che appare dopo la conclusione del documento viene semplicemente ignorato.

Subito dopo l'apertura del documento viene collocata normalmente l'istruzione di creazione del titolo, '`\maketitle`', seguita eventualmente dal comando di creazione dell'indice, '`\tableofcontents`':

```
\begin{document}
\maketitle
\tableofcontents
...
...
\end{document}
```

Segue il riepilogo dei comandi che riguardano in particolare l'inserimento dell'indice generale e di quelli specifici di figure e tabelle:

Comando	Descrizione
<code>\maketitle</code>	genera la pagina del titolo, traendo i dati dai comandi ' <code>\title</code> ', ' <code>\author</code> ', ' <code>\date</code> ' e ' <code>\thanks</code> ' usati nel preambolo;
<code>\tableofcontents</code>	inserisce l'indice generale del documento;
<code>\listoffigures</code>	elenca le figure che appaiono nel documento;
<code>\listoftables</code>	elenca le tabelle che appaiono nel documento.

A seconda della classe del documento, i comandi elencati nella tabella, che servono per la creazione di un indice, prevedono l’inserimento di un titolo automatico, attraverso un comando di dichiarazione di un capitolo o di una sezione non numerati (con un asterisco, come spiegato nella sezione successiva).

Si osservi che, a differenza dell’indice generale, la creazione di un indice analitico è un processo più complicato, descritto nella sezione [50.10](#).

50.5.4 Suddivisione del documento

«

Normalmente, il corpo del documento può essere suddiviso, a seconda del tipo di classe utilizzato. Si usano per questo i comandi elencati nella tabella seguente (anche l’asterisco fa parte dell’istruzione):

Comando	Descrizione
<code>\part [opzioni] {titolo_della_parte}</code> <code>\part* [opzioni] {titolo_della_parte}</code>	inizia una parte (solo per le classi ‘ article ’, ‘ report ’ e ‘ book ’);
<code>\chapter [opzioni] {titolo_del_capitolo}</code> <code>\chapter* [opzioni] {titolo_del_capitolo}</code>	inizia un capitolo (solo per le classi ‘ report ’ e ‘ book ’);
<code>\section [opzioni] {titolo_della_sezione}</code> <code>\section* [opzioni] {titolo_della_sezione}</code>	inizia una sezione;

Comando	Descrizione
\backslash subsection[<i>opzioni</i>] \leftarrow \hookrightarrow { <i>titolo_della_sottosezione</i> } \backslash subsection*[<i>opzioni</i>] \leftarrow \hookrightarrow { <i>titolo_della_sottosezione</i> }	inizia una sottosezione;
\backslash subsubsection[<i>opzioni</i>] \leftarrow \hookrightarrow { <i>titolo_della_sotto_sottosezione</i> } \backslash subsubsection*[<i>opzioni</i>] \leftarrow \hookrightarrow { <i>titolo_della_sotto_sottosezione</i> }	inizia una sotto-sottosezione;
\backslash paragraph[<i>opzioni</i>] { <i>titolo_del_paragrafo</i> } \backslash paragraph*[<i>opzioni</i>] { <i>titolo_del_paragrafo</i> }	inizia un paragrafo;
\backslash subparagraph[<i>opzioni</i>] \leftarrow \hookrightarrow { <i>titolo_del_sottoparagrafo</i> } \backslash subparagraph*[<i>opzioni</i>] \leftarrow \hookrightarrow { <i>titolo_del_sottoparagrafo</i> }	inizia un sottoparagrafo;
\backslash appendix	definisce l'inizio delle appendici.

In pratica, tutti questi comandi, escluso ‘ \backslash **appendix**’, dichiarano un titolo, che a seconda del caso appartiene all'inizio di una parte, di un capitolo o di una suddivisione inferiore. Come si vede dai modelli sintattici, questi comandi prevedono la presenza di opzioni, che però in pratica non si usano; inoltre, esistono due versioni del comando, con o senza asterisco.

I comandi da ‘ \backslash **part**’ a ‘ \backslash **subsection**’, vengono utilizzati per la costruzione dell'indice generale (che si può poi ottenere con il co-

mando ‘`\tableofcontents`’); inoltre, questi comandi prevedono la numerazione automatica delle voci, a seconda del livello di importanza. Se si usano questi comandi con l’asterisco, viene a mancare la numerazione e l’inserimento nell’indice generale. Le suddivisioni di livello inferiore non prevedono la numerazione e l’inserimento nell’indice, pertanto l’asterisco è praticamente privo di effetto.

```
\documentclass{report}
...
\begin{document}

\chapter{\LaTeX}

Bla bla bla bla bla bla bla...

\section{Suddivisione del documento}

Bla bla bla bla bla bla bla...

\section*{Sezione non numerata}

Questa sezione non è numerata e non ha una voce nell’indice
generale.

\subsection{Sottosezioni}

Questa è una sottosezione normale, bla bla bla...

\subsubsection{Sotto-sottosezione}

Questa è una sotto-sottosezione normale, bla bla bla...

\paragraph{Paragrafo normale}
```

```
Questo è un paragrafo normale, bla bla bla...

\subparagraph{Sottoparagrafo normale}

Questo è un sottoparagrafo normale, bla bla bla...

\end{document}
```

L'esempio, in cui non appare il preambolo (nel quale si deve definire l'uso della lingua italiana, la codifica adatta e una classe del documento che consenta la suddivisione in capitoli), si traduce nel risultato visibile nella figura 50.69, dove si può osservare in che modo procede la numerazione delle sezioni e l'aspetto normale delle suddivisioni di importanza minore.

Figura 50.69. Esempio nell'uso dei comandi di suddivisione del documento.

Capitolo 1

L^AT_EX

Bla bla bla bla bla bla bla...

1.1 Suddivisione del documento

Bla bla bla bla bla bla bla...

Sezione non numerata

Questa sezione non è numerata e non ha una voce nell'indice generale.

1.1.1 Sottosezioni

Questa è una sottosezione normale, bla bla bla...

Sotto-sottosezione

Questa è una sotto-sottosezione normale, bla bla bla...

Paragrafo normale Questo è un paragrafo normale, bla bla bla...

Sottoparagrafo normale Questo è un sottoparagrafo normale, bla bla bla...

Quando si incontra il comando ‘**\appendix**’, la numerazione dei capitoli cambia, diventando alfabetica. Se nell’esempio già mostrato si inserisce il comando ‘**\appendix**’ prima della dichiarazione del capitolo, si ottiene il risultato che vi vede nella figura 50.71

```
\appendix  
  
\chapter{\LaTeX}  
  
Bla bla bla bla bla bla bla...  
  
...
```

Figura 50.71. Appendici.

Appendice A

L^AT_EX

Bla bla bla bla bla bla bla...

A.1 Suddivisione del documento

Bla bla bla bla bla bla bla...

Sezione non numerata

Questa sezione non è numerata e non ha una voce nell'indice generale.

A.1.1 Sottosezioni

Questa è una sottosezione normale, bla bla bla...

Sotto-sottosezione

Questa è una sotto-sottosezione normale, bla bla bla...

Paragrafo normale Questo è un paragrafo normale, bla bla bla...

Sottoparagrafo normale Questo è un sottoparagrafo normale, bla bla bla...

50.5.5 Ambienti

LaTeX introduce un tipo nuovo di comando che serve a definire degli *ambienti*. L'istruzione '`\begin{ambiente}`' delimita l'inizio di un ambiente le cui caratteristiche sono definite dal nome contenuto tra le parentesi graffe; l'istruzione '`\end{ambiente}`' delimita la fine dell'ambiente dichiarato in precedenza. Per esempio, l'ambiente '`document`' definisce la zona in cui appare il corpo del documento.

A titolo di esempio, vengono elencati alcuni ambienti comuni, utilizzati frequentemente per definire le caratteristiche di paragrafi particolari:

Comando	Descrizione
<code>\begin{center}testo\end{center}</code>	centra orizzontalmente il testo;
<code>\begin{quote}testo\end{quote}</code> <code>\begin{quotation}testo\end{quotation}</code>	fa rientrare leggermente il testo a destra e a sinistra;
<code>\begin{verbatim}testo\end{verbatim}</code>	mantiene il testo esattamente come digitato nel sorgente.

L'esempio seguente mostra i comandi necessari a centrare il testo «Ciao a tutti!».

```
\begin{center}
Ciao a tutti!
\end{center}
```

50.6 LaTeX: la pagina



La pagina di LaTeX viene definita automaticamente, in base alle opzioni usate nella definizione della classe del documento; eventualmente è possibile intervenire per cambiare l'impostazione predefinita, ma in tal caso tutto diventa piuttosto complesso.

50.6.1 Dimensioni della pagina



Le dimensioni della pagina sono definite da una serie di lunghezze rigide che possono essere modificate facilmente, ma per questo occorre conoscere le varie competenze di queste variabili. In particolare, occorre tenere presente che, a seconda della classe, ci può essere o meno una distinzione tra pagine destre e sinistre.

Figura 50.74a. Lunghezze che definiscono la forma della pagina:
pagina sinistra, o pagina pari.

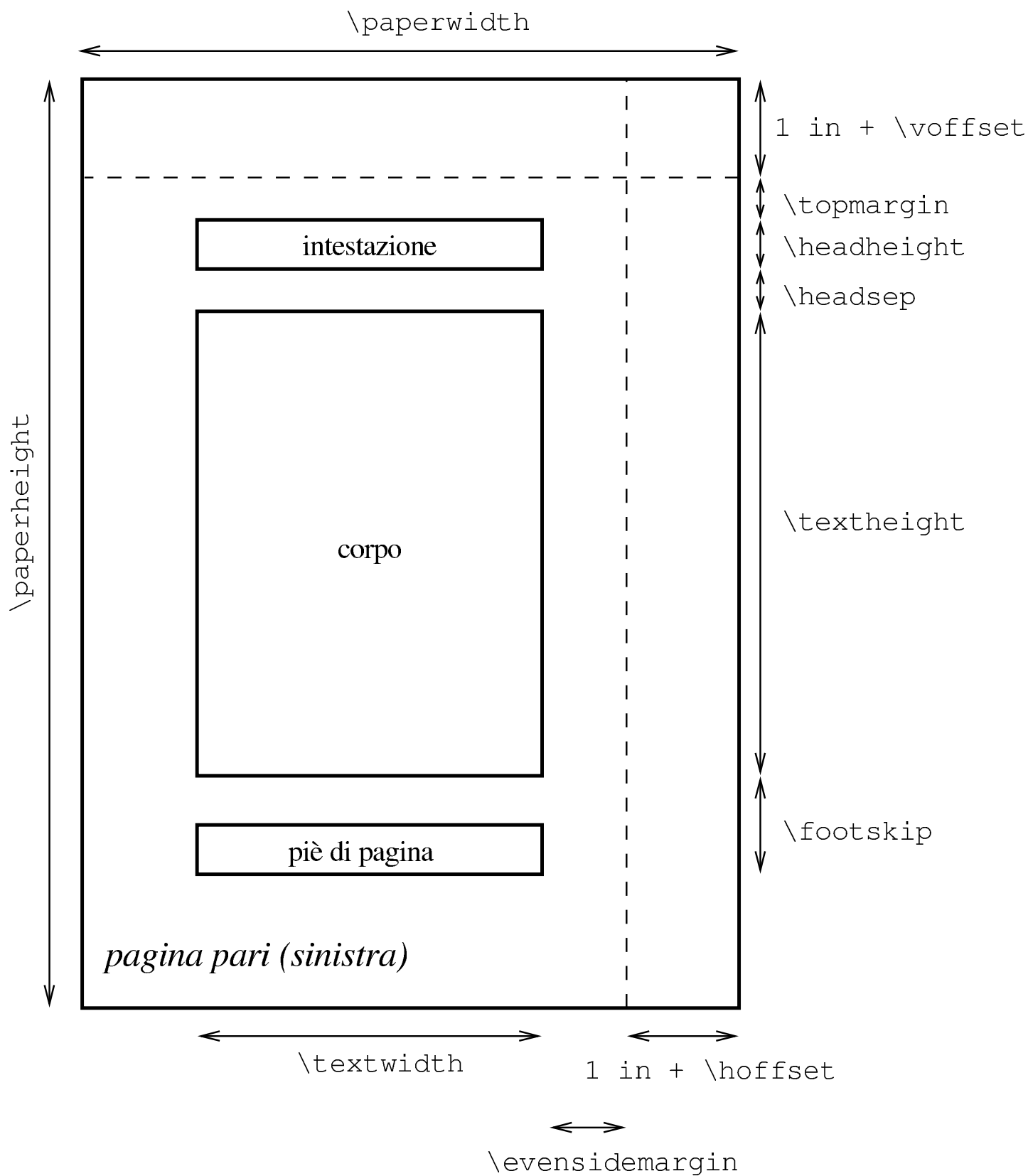
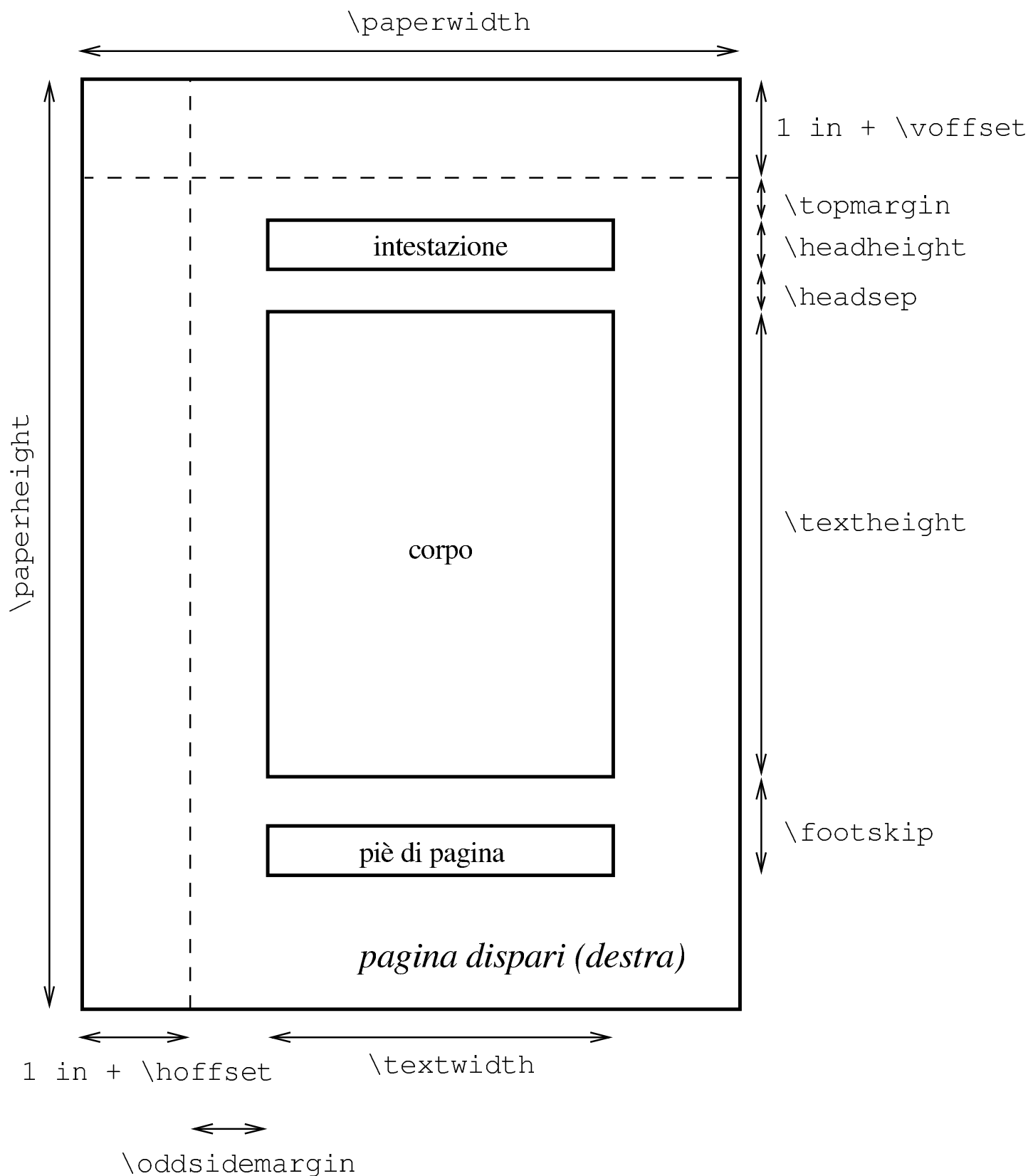


Figura 50.74b. Lunghezze che definiscono la forma della pagina:
la pagina destra o pagina dispari.



La figura 50.74 mostra schematicamente la competenza delle varie lunghezze utilizzate; in particolare, si osservi che si fa riferimento sempre a pagine destre, salvo il caso di classi che distinguono tra pagine destre e sinistre. La tabella 50.76 descrive le varie lunghezze, mostrando il valore predefinito in presenza della classe ‘**book**’, con l’opzione ‘**a4paper**’.

Tabella 50.76. Elenco dei comandi che intervengono nella definizione delle dimensioni della pagina.

Lunghezza	Valore predefinito	Descrizione
<code>\paperwidth</code>	614.295pt	Larghezza del foglio di carta su cui si stampa.
<code>\paperheight</code>	794.96999pt	Altezza del foglio di carta su cui si stampa.
<code>\hoffset</code>	0.0pt	Scostamento orizzontale dal bordo interno, dopo il margine di 2,54 cm (1 in).
<code>\voffset</code>	0.0pt	Scostamento verticale dal bordo superiore, dopo il margine di 2,54 cm (1 in).
<code>\oddsidemargin</code>	35.0pt	Margine dopo lo scostamento orizzontale, per le pagine destre.
<code>\evensidemargin</code>	89.0pt	Margine dopo lo scostamento orizzontale, per le pagine sinistre.
<code>\textwidth</code>	398.33858pt	Larghezza del corpo in cui si inserisce il testo.
<code>\textheight</code>	550.0pt	Altezza del corpo in cui si inserisce il testo.
<code>\columnwidth</code>	398.33858pt	Larghezza della colonna (se la pagina contiene una sola colonna di testo, la larghezza corrisponde a quella del corpo a disposizione).
<code>\columnsep</code>	10.0pt	Distanza tra le colonne, quando la composizione è a due colonne.
<code>\headheight</code>	12.0pt	Altezza dell'intestazione.
<code>\topmargin</code>	22.0pt	Margine dopo lo scostamento verticale, fino alla parte superiore della riga di intestazione.
<code>\headsep</code>	18.06749pt	Spazio di separazione verticale tra la base della riga di intestazione e la parte superiore del corpo.
<code>\footskip</code>	25.29494pt	Spazio tra la parte inferiore del corpo e la base della riga del piè di pagina.

È importante osservare i margini (`\topmargin`, `\oddsidemargin` o `\evensidemargin`) fanno riferimento a un punto di inizio spostato di un pollice, sia dal margine interno, sia dall'alto; pertanto, se si vogliono avere margini inferiori a 2,54 cm, i valori delle lunghezze di scostamento relative possono essere negativi.

50.6.2 Flusso verticale del testo, colonne e salti pagina

Con LaTeX, il testo viene distribuito verticalmente in modo analogo a quanto fa già TeX; tuttavia, cambiano i comandi per interrompere le righe e per il salto pagina, inoltre si inseriscono comandi per richiedere di non saltare pagina in un certo punto. La tabella seguente elenca i comandi che consentono un controllo del salto pagina:

Comando	Descrizione
<code>\newline</code>	interrompe il flusso del blocco per iniziare una riga nuova;
<code>\linebreak[0 1 2 3 4]</code>	richiede un'interruzione di riga con un grado di intensità pari al numero tra parentesi quadre (il numero maggiore richiede l'interruzione di riga con maggiore insistenza);
<code>\newpage</code>	salta all'inizio della prossima colonna;
<code>\clearpage</code>	esegue un salto pagina, assicurando che vengano inseriti gli oggetti fluttuanti che non hanno ancora trovato posto nella composizione;

Comando	Descrizione
<code>\cleardoublepage</code>	esegue un salto pagina, in modo da iniziare con una pagina destra (dispari), assicurando che vengano inseriti gli oggetti fluttuanti che non hanno ancora trovato posto nella composizione;
<code>\pagebreak [0 1 2 3 4]</code>	richiede un salto pagina con un grado di intensità pari al numero tra parentesi quadre (il numero maggiore richiede il salto pagina con maggiore insistenza);
<code>\nopagebreak [0 1 2 3 4]</code>	richiede che non venga fatto un salto pagina nel punto in cui appare il comando, con un grado di intensità pari al numero tra parentesi quadre (il numero maggiore richiede che non si faccia il salto pagina con maggiore insistenza);
<code>{\samepage <i>testo</i>}</code>	richiede che il testo indicato rimanga nella stessa pagina;
<code>\vfill</code>	inserisce uno spazio verticale elastico.

I comandi ‘`\newline`’, ‘`\newpage`’, ‘`\clearpage`’ e ‘`\cleardoublepage`’ sono drastici (il particolare legato agli oggetti fluttuanti viene chiarito nel capitolo in cui si descrive l’inserimento di immagini e di tabelle fluttuanti), mentre ‘`\linebreak`’, ‘`\nolinebreak`’, ‘`\pagebreak`’ e ‘`\nopagebreak`’, con o senza opzione, dipendono dalla calcolo di convenienza che viene fatto in fase di composizione.

Per tenere unita una porzione di testo, il metodo più efficace è dato dal comando ‘`\samepage`’, che deve essere inserito in un raggruppamento tra parentesi graffe, come si vede nella tabella riepilogativa. Tuttavia, anche in questo caso, non si può garantire che tutto il pezzo richiesto rimanga nella stessa pagina: la probabilità che sia inserita un’interruzione di pagina aumenta con l’allontanarsi dal comando.

Tra questi comandi, ‘`\vfill`’ non serve per ottenere o per evitare dei salti pagina, ma consente di spostare in basso del testo, quando viene usato con la richiesta di un salto pagina. Si osservi l’esempio seguente:

```
\noindent Bla bla bla bla bla bla...
```

```
\vfill
```

```
\noindent Copyright © 2002–2022 Tizio Tizi
```

```
\noindent This work is free; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version
2 of the License, or (at your option) any later version.
```

```
\noindent This work is distributed in the hope that it will
be useful, but WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. See the GNU General Public License for
more details.
```

```
\noindent You should have received a copy of the GNU General
Public License along with this work; if not, write to the
Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA
02139, USA.
```

```
\newpage
```

La figura 50.79 mostra il risultato della composizione, simulando una pagina molto corta, per questioni di spazio.

Figura 50.79. Allineamento in basso del testo con l'aiuto di `'\vfill'`.

Bla bla bla bla bla bla...

Copyright © 2002-2022 Tizio Tizi

This work is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This work is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this work; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

È da osservare che, a differenza del linguaggio TeX puro e semplice, i comandi per il salto pagina provvedono da soli ad aggiungere uno spazio elastico adeguato, per evitare che il contenuto della pagina venga allungato in modo inopportuno.

I comandi che richiedono un salto pagina, non si possono cumulare. In pratica, due comandi ‘`\newpage`’ in sequenza, generano un salto solo. Per riuscire a ottenere un salto pagina aggiuntivo, occorre scrivere qualcosa, come uno spazio non interrompibile, prima di un nuovo salto pagina. È per questo che esiste il comando ‘`\cleardoublepage`’ che permette di raggiungere l’inizio di una pagina destra.

LaTeX consente di controllare in modo complessivo l’allineamento verticale, attraverso due comandi molto semplici:

Comando	Descrizione
<code>\raggedbottom</code>	lascia che il testo si distribuisca verticalmente in modo naturale, senza tentare di allungarlo per occupare tutto lo spazio a disposizione;
<code>\flushbottom</code>	se possibile, allunga verticalmente gli spazi per occupare tutto lo spazio verticale della pagina.

LaTeX utilizza in modo predefinito una o l’altra forma, a seconda della classe del documento. Come già accennato, i comandi per il salto pagina inseriscono automaticamente uno spazio elastico sufficiente a evitare che in quel punto si tenti di allungare il testo fino alla fine della pagina; tuttavia, ci sono situazioni in cui con la modalità corrispondente al comando ‘`\flushbottom`’ si ottengono pagine troppo allungate.

In condizioni normali, LaTeX genera una composizione su una sola colonna verticale; tuttavia è possibile distribuire il testo su due colonne, con l’uso di due comandi particolari, che consentono di

cambiare da una modalità all'altra:

Comando	Descrizione
<code>\onecolumn</code>	inizia una nuova pagina con una sola colonna;
<code>\twocolumn</code>	inizia una nuova pagina con due colonne;
<code>\twocolumn[<i>testo</i>]</code>	inizia una nuova pagina con due colonne, mettendo il testo che appare tra parentesi quadre all'inizio, ma distribuito orizzontalmente su entrambe le colonne.

L'esempio seguente consente di vedere in pratica una composizione a due colonne. Si può osservare il fatto che la frase iniziale viene posta a cavallo delle due colonne.

```
\twocolumn[Bla bla bla bla... questo testo va a cavallo
delle due colonne]
```

```
Qui inizia la suddivisione del testo in due colonne
verticali. Bla bla bla...
```

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla...
```

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla...
```

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla...
```

```
\newpage
```

```
Qui inizia un'altra colonna, a seguito dell'uso del comando
di salto pagina semplice. Bla bla bla bla bla bla bla bla
```

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Come si può osservare nella composizione finale che appare nella figura 50.83, il comando `'\newpage'` genera un salto fino all'inizio della colonna successiva, indipendentemente dal fatto che questa si trovi nella pagina corrente o in quella successiva. Al contrario, i comandi `'\clearpage'` e, necessariamente, `'cleardoublepage'`, fanno ricominciare la composizione effettivamente in una pagina nuova.

Figura 50.83. Esempio di composizione a due colonne.

Bla bla bla bla... questo testo va a cavallo delle due colonne

Qui inizia la suddivisione del testo in due colonne verticali. Bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Qui inizia un'altra colonna, a seguito dell'uso del comando di salto pagina semplice. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Il comando `'\twocolumn'` crea due colonne di larghezza uguale, utilizzando lo spazio definito dalla lunghezza `'\textwidth'`, meno lo spazio di `'\columnsep'`. Pertanto, se si desidera cambiare la larghezza dello spazio interno alle colonne occorre intervenire prima nella lunghezza `'\columnsep'` e quindi usare il comando `'\twocolumn'`.

50.6.3 Stile della pagina



LaTeX definisce in modo automatico l'aspetto della riga di intestazione e di quella del fondo pagina, in base alla classe del documento, inserendo eventualmente delle varianti all'inizio delle parti e dei capitoli. I comandi elencati nella tabella seguente, consentono di interferire nello stile della pagina:

Comando	Descrizione
<code>\pagestyle{plain}</code> <code>\thispagestyle{plain}</code>	richiede lo stile più semplice, in cui viene mostrato soltanto il numero di pagina;
<code>\pagestyle{empty}</code> <code>\thispagestyle{empty}</code>	richiede che venga eliminata qualunque indicazione, sopra e sotto il corpo del testo;
<code>\pagestyle{headings}</code> <code>\thispagestyle{headings}</code>	richiede che si utilizzino intestazioni e righe in fondo alla pagina in modo predefinito, secondo la classe del documento;
<code>\pagestyle{myheadings}</code> <code>\thispagestyle{myheadings}</code>	richiede che si utilizzino intestazioni e righe in fondo alla pagina secondo quanto richiesto espressamente con i comandi <code>\markboth</code> o <code>\markright</code> ;

Comando	Descrizione
$\backslash\text{markboth}\{intestazione_sinistra\} \leftarrow$ $\rightarrow \{intestazione_destra\}$	definisce delle righe di intestazione personalizzate, distinte per le pagine sinistre e per le pagine destre;
$\backslash\text{markright}\{intestazione_destra\}$	definisce la riga di intestazione personalizzata per le pagine destre (si usa quando la classe del documento prevede di non distinguere tra pagine sinistre e destre);

Nell'elenco, sono stati messi assieme i comandi ' **$\backslash\text{pagemstyle}$** ' e ' **$\backslash\text{thispagemstyle}$** ', ma il primo interviene su tutte le pagine, a partire da quella in corso, fino al prossimo comando del genere, mentre il secondo riguarda esclusivamente la pagina in corso, come eccezione rispetto allo stile normale. LaTeX utilizza proprio questa tecnica per fare in modo che la pagina iniziale di un capitolo o di una parte sia diversa nello stile rispetto alle altre pagine normali.

Generalmente, le righe di intestazione sono definite automaticamente in modo appropriato, secondo la classe prescelta del documento. Per modificare manualmente queste righe di intestazione, si deve definire lo stile personalizzato, con il comando ' **$\backslash\text{pagemstyle}\{\text{myheadings}\}$** ', specificando il contenuto delle intestazioni con ' **$\backslash\text{markboth}$** ' o con ' **$\backslash\text{markright}$** '. La definizione personalizzata delle intestazioni si limita a una riga di testo molto semplice, che viene allineata in modo appropriato (a sinistra nelle pagine destre e a destra nelle pagine sinistre), per lasciare il posto

al numero della pagina che viene inserito da LaTeX, senza poter intervenire su questo particolare.

A proposito dell'uso di `\markboth` e di `\markright`, è interessante osservare che, se ci si trova in una pagina sinistra, viene preso in considerazione l'ultimo comando `\markboth` che rientri nella pagina, mentre se ci si trova in una pagina destra, viene preso in considerazione il primo comando `\markboth` o `\markright` che si trovi nella pagina stessa.

50.6.4 Numerazione delle pagine

«

LaTeX gestisce la numerazione delle pagine in modo automatico, attraverso il contatore `page`. A seconda della classe del documento, il contatore viene inizializzato in certi punti chiave, mostrando eventualmente una sequenza di numerazione differente. Se necessario, è possibile intervenire sul contatore e sull'aspetto della sequenza di numerazione.

Comando	Descrizione
<code>\pagenumbering{arabic}</code>	richiede una sequenza numerica normale (numeri arabi);
<code>\pagenumbering{roman}</code>	richiede una sequenza numerica con numeri romani fatti di lettere minuscole;
<code>\pagenumbering{Roman}</code>	richiede una sequenza numerica con numeri romani fatti di lettere maiuscole;
<code>\pagenumbering{alph}</code>	richiede una sequenza alfabetica minuscola;
<code>\pagenumbering{Alph}</code>	richiede una sequenza alfabetica maiuscola;

Comando	Descrizione
<code>\setcounter{page}{<i>n</i>}</code>	assegna alla pagina corrente il numero indicato; anche se si assegna un numero negativo, la progressione rimane in senso positivo.

Nella sezione [50.4.2](#) è stato descritto l'uso dei contatori; per modificare il contatore della pagina, è sufficiente agire sul nome '**page**', per esempio reinizializzandolo in questo modo:

```
\setcounter{page}{1}
```

Naturalmente, per intervenire in questo modo ci deve essere un buon motivo, perché la classe del documento provvede già a definire il modo migliore per reinizializzare il contatore al momento più opportuno.

Nello stesso modo, potrebbe essere interessante cambiare il modo in cui viene mostrato il numero della pagina, attraverso il comando '**\pagenumbering**', come descritto nella tabella.

50.6.5 Note a piè di pagina

Le note a piè di pagina vengono definite normalmente attraverso il comando '**\footnote**', che provvede da solo a incrementare il contatore delle note, corrispondente a '**footnote**', e a inserire la nota nella collocazione appropriata. Tuttavia, volendo fare riferimento alla stessa nota in più punti distinti, occorre separare la fase di dichiarazione da quella in cui si definisce il testo associato.



Comando	Descrizione
<code>\footnote{<i>testo</i>}</code>	incrementa il contatore delle note, inserisce il numero ad apice nel testo e mette la nota a piè di pagina;
<code>\footnotemark[\value{footnote}]</code>	lascia inalterato il contatore delle note e inserisce il numero attuale del contatore ad apice nel testo;
<code>\footnotemark[<i>n</i>]</code>	lascia inalterato il contatore delle note e inserisce il numero <i>n</i> ad apice nel testo;
<code>\footnotemark</code>	incrementa il contatore delle note, inserisce il numero ad apice nel testo, ma per la nota a piè di pagina attende il comando ' <code>\footnotetext</code> ';
<code>\footnotetext{<i>testo</i>}</code>	mette la nota a piè di pagina, associandole il numero attuale del contatore ' <code>footnote</code> '.
<code>\footnotetext[<i>n</i>]{<i>testo</i>}</code>	mette la nota a piè di pagina, associandole il numero <i>n</i> .

50.6.6 Sillabazione

Come per TeX, si può usare il comando ‘`\hyphenation`’ per dichiarare la sillabazione di parole che in qualche modo non vengono separate correttamente; per esempio così:

```
\hyphenation{car-tel-lo-ne ma-nu-a-le}
```

Eventualmente, come già annotato in un’altra sezione, se c’è la necessità di mantenere unita una parola, o più parole assieme, si può usare una scatola:

```
\mbox{file system}
```

In questo caso, oltre che impedire la separazione delle parole, si vuole anche evitare che la riga venga interrotta in corrispondenza dello spazio che le separa.

50.7 Caratteri

In generale, LaTeX eredita la gestione dei caratteri da stampa del linguaggio TeX puro e semplice. In questo senso, valgono praticamente le stesse regole per ottenere le lettere accentate e gli altri simboli speciali, con la differenza che è possibile istruire LaTeX in modo che accetti la codifica UTF-8, con le istruzioni seguenti, da inserire nel preambolo, come descritto nella sezione [50.1](#).

```
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
```

Anche con LaTeX funzionano i legati in modo automatico, inoltre si definiscono i contesti matematici nello stesso modo di TeX.

La differenza importante di LaTeX sta nella definizione di una veste grafica complessiva e nel modo di fare riferimento ai ca-

ratteri da stampa, che avviene attraverso comandi differenti, con comportamenti diversi rispetto a TeX.

Negli esempi, salvo indicazione diversa, si suppone di utilizzare nel preambolo i comandi necessari all'utilizzo della codifica UTF-8:

```
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
```

50.7.1 Caratteri che hanno un significato speciale

«

Alcuni simboli (caratteri) usati nel sorgente hanno necessariamente un significato speciale, nel senso che non si traducono nel segno corrispondente nella composizione. Questo problema esiste già con TeX, ma LaTeX offre delle alternative leggermente diverse nel modo di ottenere i simboli mancanti nella composizione. La tabella 50.92 riepiloga i simboli che hanno un significato speciale nel sorgente LaTeX, con l'indicazione del codice necessario per ottenere i segni corrispondenti nella composizione.

Tabella 50.92. Elenco dei caratteri che hanno significati particolari.²

Carattere speciale	Utilizzo normale	Trasformazione per l'uso letterale
\	Prefisso di un comando o di un simbolo di controllo.	\textbackslash

Carattere speciale	Utilizzo normale	Trasformazione per l'uso letterale
{ }	Apri e chiude un gruppo.	\{ \}
%	Inizia un commento fino alla fine della riga.	\%
&	Tabulazione orizzontale.	\&
~	Spazio non interrompibile.	\textasciitilde
\$	Inizia e conclude un contesto matematico.	\\$
^	Apice, in un contesto matematico.	\textasciicircum
_	Il trattino basso rappresenta un pedice, in un contesto matematico.	_{}
#	Definisce la collocazione di un parametro.	\#
<SP>	Uno o più spazi vengono ridotti a uno solo.	\<SP>
[]	Alle volte, le parentesi quadre delimitano un'opzione e non vengono inserite nella composizione.	{ [] { [] }

Per quanto riguarda gli accenti ottenuti sovrapponendo il simbolo alla lettera successiva, il funzionamento è uguale a quello di TeX, come descritto nelle sezioni [49.10.2](#) e [49.17.1](#).

50.7.2 Corpi uniformi



Lo scopo principale di LaTeX è quello di impostare una veste grafica uniforme, dove, tra le altre cose si specificano inizialmente i corpi e gli stili dei caratteri da stampa da usare nei vari contesti. In particolare, con il comando iniziale ‘`\documentclass`’ è possibile indicare la dimensione normale del testo, attraverso un’opzione (come ‘`11pt`’), a cui si adegua proporzionalmente tutto il resto (sezione [50.1](#)).

Alcuni comandi definiscono il corpo del carattere in base a delle definizioni relative:

Comando	Descrizione
<pre>{\Huge <i>testo</i> } \begin{Huge}<i>testo</i>\end{Huge}</pre>	il corpo più grande della serie;
<pre>{\huge <i>testo</i> } \begin{huge}<i>testo</i>\end{huge}</pre>	
<pre>{\LARGE <i>testo</i> } \begin{LARGE}<i>testo</i>\end{LARGE}</pre>	
<pre>{\Large <i>testo</i> } \begin{Large}<i>testo</i>\end{Large}</pre>	
<pre>{\large <i>testo</i> } \begin{large}<i>testo</i>\end{large}</pre>	

Comando	Descrizione
<pre>{\normalsize <i>testo</i> } \begin{normalsize} <i>testo</i> \end{normalsize}</pre>	<p>la dimensione normale del testo;</p>
<pre>{\small <i>testo</i> } \begin{small}<i>testo</i>\end{small}</pre>	
<pre>{\footnotesize <i>testo</i> } \begin{footnotesize} <i>testo</i> \end{footnotesize}</pre>	
<pre>{\scriptsize <i>testo</i> } \begin{scriptsize} <i>testo</i> \end{scriptsize}</pre>	
<pre>{\tiny <i>testo</i> } \begin{tiny}<i>testo</i>\end{tiny}</pre>	<p>il corpo più piccolo nell'ambito della veste grafica scelta.</p>

In condizioni normali, questi comandi non vengono usati direttamente da chi scrive un documento in LaTeX; tuttavia, se c'è la necessità di ingrandire o di rimpicciolire il testo, con questi c'è la certezza di rimanere nelle dimensioni standard usate dallo stile prescelto:

Questo è un testo normale, che può diventare più `{\large grande}` o più `{\small piccolo}`, a seconda delle esigenze.

Questo è un testo normale, che può diventare più grande o più piccolo, a seconda delle esigenze.

50.7.3 Caratteri da stampa

«

A differenza di TeX, i comandi di LaTeX che definiscono lo stile del carattere sono cumulabili. Per esempio, la definizione di un testo in corsivo può essere cumulata con quella del neretto, per ottenere un neretto corsivo.

Le caratteristiche principali di un carattere sono lo stile, la serie e il corpo. Lo stile e la serie possono essere definite utilizzando istruzioni del tipo seguente:

1. `\stile_serie_a {testo}`

2. `{\stile_serie_b testo}`

3. `\begin{stile_serie_b}...\end{stile_serie_b}`

I nomi delle istruzioni cambiano a seconda che si utilizzi un modo oppure l'altro. Per esempio, per scrivere un testo in corsivo, si possono utilizzare questi tre modi.

```
\textit{Testo in corsivo}

{\itshape Testo in corsivo}

\begin{itshape}Testo in corsivo\end{itshape}
```

Tutte le forme consentono di cumulare gli effetti; per esempio, si può scrivere ‘`\textit{\textbf{testo}}`’, ‘`{\itshape\bfseries testo}`’, oppure ‘`\begin{itshape}\begin{bfseries}testo\end{bfseries}\end{itshape}`’ per ottenere un neretto corsivo.

La tabella seguente elenca i comandi che possono essere usati in un contesto normale (non matematico):

Comando	Descrizione
<pre>\textrm{<i>testo</i>} {\rmfamily <i>testo</i>} \begin{rmfamily}<i>testo</i>\end{rmfamily}</pre>	tondo;
<pre>\textsf{<i>testo</i>} {\sffamily <i>testo</i>} \begin{sffamily}<i>testo</i>\end{sffamily}</pre>	lineare;
<pre>\texttt{<i>testo</i>} {\ttfamily <i>testo</i>} \begin{ttfamily}<i>testo</i>\end{ttfamily}</pre>	dattilografico;

Comando	Descrizione
<pre>\textbf{<i>testo</i>}</pre> <pre>{\bfseries <i>testo</i>}</pre> <pre>\begin{bfseries}<i>testo</i>\end{bfseries}</pre>	neretto;
<pre>\textup{<i>testo</i>}</pre> <pre>{\upshape <i>testo</i>}</pre> <pre>\begin{upshape}<i>testo</i>\end{upshape}</pre>	normale, in contrapposizione al neretto;
<pre>\textit{<i>testo</i>}</pre> <pre>{\itshape <i>testo</i>}</pre> <pre>\begin{itshape}<i>testo</i>\end{itshape}</pre>	corsivo;
<pre>\textsl{<i>testo</i>}</pre> <pre>{\slshape <i>testo</i>}</pre> <pre>\begin{slshape}<i>testo</i>\end{slshape}</pre>	inclinato;
<pre>\textsc{<i>testo</i>}</pre> <pre>{\scshape <i>testo</i>}</pre> <pre>\begin{scshape}<i>testo</i>\end{scshape}</pre>	maiuscoletto;

Comando	Descrizione
$\text{\textnormal{testo}}$ $\{\text{\normalfont testo}\}$ $\text{\begin{normalfont}}$ <i>testo</i> $\text{\end{normalfont}}$	testo normale.

La tabella successiva riguarda invece il contesto matematico:

Comando	Descrizione
$\text{\mathnormal{testo}}$	testo matematico normale;
$\text{\mathrm{testo}}$	tondo;
$\text{\mathbf{testo}}$	neretto;
$\text{\mathsf{testo}}$	lineare;
$\text{\mathtt{testo}}$	dattilografico;
$\text{\mathit{testo}}$	corsivo;
$\text{\mathcal{testo}}$	calligrafico;
$\text{\mathsc{testo}}$	maiuscoletto.

A titolo di esempio, la tabella 50.99 consente di confrontare le varianti seriali dello stile normale di scrittura.

Tabella 50.99. Combinazioni a coppie di serie per il testo normale.

	tondo	corsivo	neretto	inclinato	maiuscoletto
tondo	abc123	<i>abc123</i>	abc123	<i>abc123</i>	ABC123
corsivo	<i>abc123</i>	<i>abc123</i>	<i>abc123</i>	<i>abc123</i>	ABC123
neretto	abc123	<i>abc123</i>	abc123	<i>abc123</i>	ABC123
inclinato	<i>abc123</i>	<i>abc123</i>	<i>abc123</i>	<i>abc123</i>	ABC123
maiuscoletto	ABC123	<i>abc123</i>	ABC123	ABC123	ABC123

Oltre allo stile e alla serie del carattere può essere definito il corpo, con i comandi già mostrati nella sezione precedente, oppure in modo diretto, nel caso si debba andare al di fuori della veste grafica preimpostata. In tal caso, si usano dei comandi nella forma:

```
\fontsize{corpo}{distanza_tra_le_righe}\selectfont
```

Per esempio, per ottenere un corpo di 5 mm, con un'altezza della riga di 6 mm (il 120 % del corpo), si può usare il comando '**\fontsize{5mm}{6mm}\selectfont**':

```
Questo è un testo normale; {\fontsize{5mm}{6mm}\selectfont
questo carattere ha un corpo particolare}; qui il testo
torna a essere normale.
```

Questo è un testo normale; questo carattere ha un corpo particolare; qui il testo torna a essere normale.

50.7.3.1 Enfaticizzazione

«

Quando non ha importanza stabilire esattamente l'aspetto di un carattere, ma si vuole semplicemente enfaticizzarlo rispetto al contesto in cui si trova, si può usare il comando '**\emph**'. L'esempio seguente mostra cosa accade quando in una situazione in cui il carattere è corsivo si usa questo comando:

```
\begin{itshape}Bla bla bla bla bla bla \emph{testo
enfaticizzato} bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla... \end{itshape}
```

Bla bla bla bla bla bla testo enfaticizzato bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

50.7.4 Definizione dettagliata del carattere da stampa

Generalmente, i comandi già descritti a proposito della definizione del carattere da stampa sono più che sufficienti per l'utente normale di LaTeX; eventualmente, sono disponibili anche altri comandi più dettagliati, che si prestano particolarmente per la realizzazione di macro particolari. Questi comandi hanno in comune la necessità di usare alla fine il comando '**\selectfont**' per attuare le modifiche al carattere da stampa. La tabella seguente elenca i comandi che riguardano la definizione dettagliata del carattere da stampa (appare anche '**\fontsize**' che è già descritto nella sezione precedente):

Comando	Descrizione
<code>\fontencoding{OT1 T1 <i>altro_insieme</i>}</code>	definisce l'insieme di caratteri (' T1 ' rappresenta quello utile per le lingue europee occidentali);
<code>\fontfamily{cmr cmss cmtt <i>altro_stile</i>}</code>	definisce lo stile del carattere (' cmr ' sta per <i>Computer modern roman</i> , ' cmss ' è il <i>Computer modern sans serifs</i> , ' cmtt ' è il <i>Computer modern typewriter text</i>);

Comando	Descrizione
<code>\fontseries{m b c bc bx <i>altra_serie</i>}</code>	definisce la serie del carattere ('m' sta per <i>Medium</i> , 'b' sta per <i>Bold</i> , 'c' sta per <i>Condensed</i> , 'bc' sta per <i>Bold condensed</i> e 'bx' sta per <i>Bold extended</i>);
<code>\fontshape{n it sl sc ui ol}</code>	definisce una variante della serie del carattere ('n' sta per <i>Normal</i> , 'it' sta per <i>Italic</i> , 'sl' sta per <i>Slanted</i> , 'sc' sta per <i>Small caps</i> , 'ui' sta per <i>Upright italic</i> e 'ol' sta per <i>Outline</i> , ma naturalmente non tutti i tipi di carattere hanno a disposizione tutte le varianti possibili);
<code>\fontsize{<i>corpo</i>}{<i>altezza_riga</i>}</code>	definisce il corpo del carattere e l'altezza della riga;
<code>\selectfont</code>	abilita le modifiche fatte in precedenza al carattere (tipo, stile e dimensione);
<code>\usefont{<i>insieme</i>}{<i>stile</i>}{<i>forma</i>}</code>	definisce simultaneamente tutte le caratteristiche del carattere, escluso il corpo e la distanza tra le righe, senza nemmeno bisogno di '\selectfont' per attivare la scelta fatta.

A proposito del comando ‘`\usefont`’, si osservi che il secondo argomento si riferisce alle sigle utilizzabili con il comando ‘`\fontseries`’, mentre il terzo argomento richiede una sigla riferita al comando ‘`\fontshape`’. La suddivisione delle varianti seriali in questi due gruppi deriva dall’organizzazione di LaTeX riguardo alla cumulabilità tra stili. In pratica, le varianti seriali di ‘`\fontseries`’ non sono cumulabili tra di loro; per esempio, per ottenere un carattere stretto e neretto è già prevista una voce unica: ‘`bc`’. Lo stesso ragionamento vale per le varianti seriali di ‘`\fontshape`’, mentre una variante di ‘`\fontseries`’ è cumulabile con una variante di ‘`\fontshape`’.

Viene mostrato un esempio completo, in cui si può notare anche l’uso delle lettere accentate, a seguito della selezione dell’insieme ‘**T1**’:

```
\documentclass{article}
\usepackage[italian]{babel}
%\usepackage[utf8x]{inputenc}
%\usepackage[T1]{fontenc}
\frenchspacing
\pagestyle{empty}
\setlength{\textwidth}{12cm}
\begin{document}
```

Questo \’e un testo normale;

```
{\fontencoding{T1}\fontfamily{cmss}\fontseries{n}
\fontshape{it}\fontsize{5mm}{7mm}\selectfont questo \’e
invece un carattere senza grazie, chiaro, corsivo, con un
corpo di 5 mm, in una riga alta 7 mm.} Questo \’e di nuovo
un carattere normale. \usefont{T1}{cmss}{n}{it}Questo \’e
ancora un carattere senza grazie, chiaro, corsivo, con un
corpo normale. \fontsize{5mm}{7mm}\selectfont Qui il
carattere si ingrandisce di nuovo. \normalfont Qui si
```

```

ritorna a un carattere di stile e serie normale, ma la
dimensione rimane inalterata.

```

```

\end{document}

```

Questo è un testo normale; questo è invece un carattere senza grazie, chiaro, corsivo, con un corpo di 5 mm, in una riga alta 7 mm. Questo è di nuovo un carattere normale. Questo è ancora un carattere senza grazie, chiaro, corsivo, con un corpo normale. Qui il carattere si ingrandisce di nuovo. Qui si ritorna a un carattere di stile e serie normale, ma la dimensione rimane inalterata.

50.7.5 Linee e cornicette attorno al testo

«

Per LaTeX vale lo stesso problema di TeX a proposito della sottolineatura, nel senso che non fa parte dello stile e della serie del carattere. Tuttavia, LaTeX facilita anche altre forme di evidenziamento analoghe:

Comando	Descrizione
<code>\underline{testo}</code>	mette una linea sotto al testo;
<code>\$\$\overline{\hbox{testo}}\$\$</code>	mette una linea sopra al testo;
<code>\frame{testo}</code>	racchiude il testo in un rettangolo;
<code>\fbox{testo}</code>	racchiude il testo in un rettangolo, lasciando un po' di spazio.

In tutti i casi, il testo evidenziato si trova inserito in una scato-

la orizzontale, che come tale non può essere spezzata, creando eventualmente dei problemi di impaginazione.

Si osservi che il comando ‘`\overline`’ è fatto per l’ambito matematico. Nella tabella, la sintassi mostra il modo corretto per poterlo utilizzare in un contesto normale, con l’aggiunta del comando ‘`\hbox`’, allo scopo di comporre il testo sovrilineato in un contesto normale.

Segue un esempio riepilogativo di queste forme di evidenziamento del testo:

```
Qui inizia un testo normale, in cui questa
\underline{parola} appare sottolineata; ma è possibile
anche sovrilineare  $\overline{\hbox{questa
piccola frase}}$ , così come si possono creare delle
\frame{cornicette schiacciate}, oppure \fbox{un po' più
distanziate} dal testo.
```

Qui inizia un testo normale, in cui questa parola appare sottolineata; ma è possibile anche sovrilineare questa piccola frase, così come si possono creare delle cornicette schiacciate, oppure un po' più distanziate dal testo.

50.7.6 Spostamento verticale del testo

A volte può essere necessario lo spostamento verticale di alcuni caratteri rispetto alla base della riga. Per fare questo si usa il comando ‘`\raisebox`’, con cui si delimita il testo da spostare, specificando di quanto questo debba essere alzato (per abbassarlo si possono usare valori negativi):



```
\raisebox{spostamento_verticale}{testo}
```

Si osservi l'esempio seguente in cui si gioca con la parola «ciao»:

```
c\raisebox{1mm}{i}\raisebox{2mm}{a}\raisebox{1mm}{o}
c\raisebox{-1mm}{i}\raisebox{-2mm}{a}\raisebox{-1mm}{o}
```

ci^ao ci_ao

50.8 LaTeX: blocchi di testo

«

LaTeX impagina il testo in paragrafi, in modo analogo a quanto fa già il linguaggio TeX puro e semplice. Anche per LaTeX funziona il comando ‘**\par**’, allo scopo di concludere un paragrafo in modo esplicito, oppure è sufficiente lasciare almeno una riga bianca tra un blocco e l’altro.

50.8.1 Caratteristiche normali di un paragrafo

«

Il paragrafo di LaTeX è costruito in maniera molto simile a quella di TeX, con la differenza che il controllo sulla sua estensione orizzontale dipende dall’impostazione della pagina e dalla suddivisione in colonne. Anche con LaTeX rimane il rientro della prima riga dei paragrafi, con l’eccezione normale del primo paragrafo dopo un titolo. La tabella 50.112 mostra alcuni dei comandi che controllano l’aspetto del paragrafo.

Tabella 50.112. Alcuni dei comandi che intervengono nella definizione del paragrafo.

Lunghezza	Valore predefinito	Descrizione
<code>\parindent</code>	0.0pt	Larghezza del rientro della prima riga (in questo caso il contesto è quello di una tabella; di solito il valore è maggiore di zero).
<code>\parskip</code>	0.0pt	Distanza aggiuntiva tra i paragrafi.
<code>\linewidth</code>	79.66649pt	Larghezza della riga (in condizioni normali coincide con la larghezza della colonna a disposizione).
<code>\baselineskip</code>	12.0pt	Distanza verticale tra la base di una riga e la base di quella successiva (questo valore viene modificato automaticamente se si cambia la dimensione del carattere da stampa).
<code>\leftskip</code>	0.0pt	Rientro sinistro del paragrafo.
<code>\rightskip</code>	0.0pt	Rientro destro del paragrafo.

Senza intervenire nella lunghezza definita dal comando `\parindent`, è possibile annullare il rientro della prima riga con il comando `\noindent`, così come è possibile usare `\indent` per ottenere l'effetto opposto in presenza di un paragrafo che non ha il rientro per qualche ragione. Tuttavia, anche annullando il rientro con il comando `\noindent`, rimane il problema della distanza tra i paragrafi, che in condizioni normali risulta troppo piccolo per consentirne l'individuazione senza rientri. Pertanto, quando si intende scrivere un documento che non fa uso di rientri in modo sistematico, conviene modificare le lunghezze che contano, direttamente nel preambolo, per esempio così:

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

Si osservi però, che in questo modo non può funzionare nemmeno il comando ‘`\indent`’, perché si limiterebbe a porre un rientro di zero punti.

Segue il riepilogo degli ultimi comandi descritti:

Comando	Descrizione
<code>\indent</code>	richiede espressamente il rientro della prima riga secondo il valore di ‘ <code>\parindent</code> ’;
<code>\noindent</code>	annulla il rientro della prima riga per il paragrafo in cui si trova.

50.8.2 Allineamento del testo

«

In condizioni normali, il testo dei paragrafi risulta allineato simultaneamente a sinistra e a destra. Per modificare questo comportamento si interviene normalmente attraverso degli ambienti, ma sono disponibili anche dei comandi alternativi che provengono da TeX:

Comando	Descrizione
<code>\begin{center} <i>testo</i> \end{center}</code>	centra il testo;
<code>\centerline{<i>riga_di_testo</i>}</code>	centra la riga;
<code>\begin{flushleft} <i>testo</i> \end{flushleft}</code> <code>{\raggedright <i>testo</i> \par}</code>	allinea il testo a sinistra;
<code>\begin{flushright}</code> <code><i>testo</i></code> <code>\end{flushright}</code> <code>{\raggedleft <i>testo</i> \par}</code>	allinea il testo a destra.

È interessante osservare che, mentre i comandi ‘`\raggedright`’ e ‘`\raggedleft`’ richiedono l’inserimento in un raggruppamento, dove l’ultimo paragrafo deve terminare in modo esplicito, prima della fine del raggruppamento, gli ambienti dichiarano implicitamente l’inizio e la fine di un paragrafo.

Il comando ‘`\centerline`’ consente di centrare una riga soltanto, senza inserire la spaziatura che riguarda un paragrafo normale.

Questo è un paragrafo normale, che serve come riferimento.
 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 bla bla bla bla.

```
\begin{center}
```

Questo è un paragrafo centrato, attraverso l’ambiente `{\ttfamily center}`. Bla bla bla bla bla bla bla bla bla bla bla.

```
\end{center}
```

Questo è un paragrafo normale, a cui segue una riga centrata:

```
\centerline{riga centrata.}
```

```
\begin{flushleft}
```

Questo è un paragrafo allineato a sinistra, attraverso l’ambiente `{\ttfamily flushleft}`. Bla bla bla bla bla bla bla bla bla bla bla.

```
\end{flushleft}
```

```
\begin{flushright}
```

Questo è un paragrafo allineato a destra, attraverso l’ambiente `{\ttfamily flushright}`. Bla bla bla bla bla bla bla bla bla bla bla.

All'interno dell'ambiente `'itemize'`, il comando `'\item'` serve a indicare l'inizio di una voce dell'elenco; la voce può contenere del testo piuttosto articolato, anche in più paragrafi, per esempio con dei sottoelenchi o ambienti di altro genere. L'esempio seguente mostra un elenco contenente un sottoelenco:

```
\begin{itemize}
\item Primo.
\item Secondo.
    \begin{itemize}
    \item Questa voce è una suddivisione ulteriore del
        secondo punto.

        Questo è un paragrafo ulteriore.
    \item Questa è una voce ulteriore della
        sottoclassificazione appartenente al secondo
        punto.
    \end{itemize}
\item Terzo.
\end{itemize}
```

- Primo.
- Secondo.
 - Questa voce è una suddivisione ulteriore del secondo punto.
Questo è un paragrafo ulteriore.
 - Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.

- Terzo.

Teoricamente, è possibile modificare il simbolo usato per evidenziare l'inizio delle voci, ridefinendo i comandi `'\labelitemi'`, `'\labelitemii'`, `'\labelitemiii'` e `'\labelitemiv'`, che si ri-

feriscono rispettivamente all'elenco principale e di seguito a sottoelenchi fino a un massimo di quattro livelli di annidamento. Per esempio, si può fare in modo che si utilizzi il segno «+» per l'elenco principale intervenendo in questo modo:

```
\renewcommand{\labelitemi}{\normalfont +}
```

Ecco come si presenta l'elenco già mostrato se si aggiunge questa istruzione all'inizio:

+ Primo.

+ Secondo.

– Questa voce è una suddivisione ulteriore del secondo punto.

Questo è un paragrafo ulteriore.

– Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.

+ Terzo.

In modo analogo si ottengono degli elenchi numerati, con l'ambiente **'enumerate'**:

```
\begin{enumerate}
\item testo
\item testo
...
\item testo
\end{enumerate}
```

L'esempio già visto viene riproposto usando l'ambiente **'enumerate'**; si può osservare che la numerazione delle voci cambia quando si inseriscono dei sottoelenchi:

```

\begin{enumerate}
\item Primo.
\item Secondo.
    \begin{enumerate}
    \item Questa voce è una suddivisione ulteriore del
        secondo punto.

        Questo è un paragrafo ulteriore.
    \item Questa è una voce ulteriore della
        sottoclassificazione appartenente al secondo
        punto.
    \end{enumerate}
\item Terzo.
\end{enumerate}

```

1. Primo.

2. Secondo.

(a) Questa voce è una suddivisione ulteriore del secondo punto.

Questo è un paragrafo ulteriore.

(b) Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.

3. Terzo.

La sequenza dei valori attribuita alle voci viene determinata attraverso dei contatori, che sono rispettivamente: ‘**enumi**’, ‘**enumii**’, ‘**enumiii**’ e ‘**enumiv**’. In pratica, il primo contatore controlla l’elenco più esterno, mentre l’ultimo riguarda quello più interno (di conseguenza ci possono essere un massimo di quattro annidamenti). Inoltre, il tipo di numerazione usata negli elenchi dipende rispettivamente dai comandi: ‘**\theenumi**’, ‘**\theenumii**’, ‘**\theenumiii**’ e ‘**\theenumiv**’. L’esempio seguente è una variante di quello già vi-

sto, in cui, l'elenco più esterno parte dal numero tre, mentre l'elenco annidato utilizza i numeri romani:

```
\begin{enumerate}
\setcounter{enumi}{2}
\item Primo.
\item Secondo.
    \renewcommand{\theenumii}{\roman{enumii}}
    \begin{enumerate}
    \item Questa voce è una suddivisione ulteriore del
        secondo punto.

        Questo è un paragrafo ulteriore.
    \item Questa è una voce ulteriore della
        sottoclassificazione appartenente al secondo
        punto.
    \end{enumerate}
\item Terzo.
\end{enumerate}
```

3. Primo.

4. Secondo.

(i) Questa voce è una suddivisione ulteriore del secondo punto.
Questo è un paragrafo ulteriore.

(ii) Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.

5. Terzo.

Gli elenchi descrittivi sono leggermente diversi, perché devono consentire la descrizione delle voci. L'ambiente usato per questo è **'description'**:


```

\begin{description}
\item [etichetta] testo
\item [etichetta] testo
...
\item [etichetta] testo
\end{description}

```

Ciò che è stato indicato come *etichetta* è ciò che rappresenta simbolicamente la voce che viene descritta dal testo successivo; come si vede dalla sintassi, potrebbe essere omessa, anche se questo fa perdere di significato a un elenco descrittivo. Viene riproposto l'esempio già visto in precedenza per gli altri tipi di elenco, con l'aggiunta delle etichette:

```

\begin{description}
\item [marrone] Primo.
\item [rosso] Secondo.
  \begin{description}
\item [oro] Questa voce è una suddivisione ulteriore
del secondo punto.

    Questo è un paragrafo ulteriore.
\item [argento] Questa è una voce ulteriore della
sottoclassificazione appartenente al secondo
punto.
  \end{description}
\item [arancio] Terzo.
\end{description}

```

marrone Primo.

rosso Secondo.

oro Questa voce è una suddivisione ulteriore del secondo punto.

Questo è un paragrafo ulteriore.

argento Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.

arancio Terzo.

Come si vede dal risultato della composizione, l'annidamento non fa cambiare l'aspetto dell'elenco.

Eventualmente, si può omettere l'indicazione della voce distintiva dei vari punti, se a questo si provvede da soli con gli evidenziamenti che si preferiscono; tuttavia, può risultare difficile inserire le parentesi quadre all'inizio di queste voci. Per inserire le parentesi quadre con un significato letterale, basta fare in modo che LaTeX comprenda che il testo è già iniziato, attraverso l'inserzione di gruppi vuoti, oppure inserendo le parentesi stesse tra parentesi graffe:

```
\begin{description}
\item [marrone] Primo.
\item {[rosso]} Secondo.
\item {}[arancio] Terzo.
\end{description}
```

marrone Primo.

[rosso] Secondo.

[arancio] Terzo.

Gli ambienti riferiti agli elenchi gestiscono in modo autonomo la distanza verticale di separazione tra le voci, impostando internamente il comando ‘\itemsep’. Volendo, si può interferire in questa lun-

ghezza per modificare la distanza tra le voci di un elenco, ma solo all'interno dell'elenco stesso:

```
\begin{itemize}
\item Primo. In questo contesto, il valore predefinito di
      {\ttfamily \textbackslash{}itemsep} è \the\itemsep{}.
\setlength{\itemsep}{1cm}
\item Secondo.
      \begin{itemize}
      \item Questa voce è una suddivisione ulteriore del
            secondo punto.

            Questo è un paragrafo ulteriore.
      \item Questa è una voce ulteriore della
            sottoclassificazione appartenente al secondo
            punto.
      \end{itemize}
\item Terzo.
\end{itemize}
```

- Primo. In questo contesto, il valore predefinito di `\itemsep` è 4.0pt plus 2.0pt minus 1.0pt.
 - Secondo.
 - Questa voce è una suddivisione ulteriore del secondo punto.
Questo è un paragrafo ulteriore.
 - Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.
 - Terzo.
- Infine, si può realizzare un elenco in modo libero, attraverso

l'ambiente `'list'`:

```
\begin{list}{etichetta}{comandi_di_spaziatura }  
\item testo  
\item testo  
...  
\item testo  
\end{list}
```

In pratica, al posto di *etichetta* si inseriscono i comandi necessari a produrre l'etichetta che contrassegna le voci dell'elenco, mentre al posto di *comandi_di_spaziatura* si possono mettere i comandi che servono a definire le lunghezze relative alla costruzione dell'elenco. Per cominciare, conviene mostrare un esempio banale di elenco puntato, dove il punto delle voci è rappresentato da un asterisco:

```
\begin{list}{*}{}  
\item Primo.  
\item Secondo.  
\item Terzo.  
\end{list}
```

* Primo.

* Secondo.

* Terzo.

Per controllare la distanza tra i punti, si può modificare la distanza `'\itemsep'`:

```
\begin{list}{*}{\setlength{\itemsep}{1cm}}  
\item Primo.  
\item Secondo.  
\item Terzo.  
\end{list}
```

* Primo.

* Secondo.

* Terzo.

Per costruire un elenco numerato, occorre intervenire con maggiore abilità nella definizione dell'etichetta:

```
\newcounter{elenco}  
  
\setcounter{elenco}{0}  
\begin{list}{\stepcounter{elenco}\arabic{elenco}}{\setlength{\itemsep}{1cm}}  
\item Primo.  
\item Secondo.  
\item Terzo.  
\end{list}
```

1 Primo.

2 Secondo.

3 Terzo.

50.8.4 Citazioni



Sono disponibili due ambienti per rappresentare facilmente del testo rientrato rispetto al flusso normale:

Comando	Descrizione
<code>\begin{quote}<i>testo</i>\end{quote}</code>	rientra a destra e a sinistra il testo, in un modo conveniente per includere una citazione.
<code>\begin{quotation}<i>testo</i>\end{quotation}</code>	

La differenza tra i due ambienti sta nel fatto che ‘**quote**’ non fa rientrare la prima riga dei paragrafi, mentre ‘**quotation**’ si comporta come al solito. In ogni caso, entrambi gli ambienti possono contenere più paragrafi o blocchi di altro tipo. Segue un esempio:

```
Bla bla bla bla bla...
```

```
\begin{quote}
```

```

This program is free software; you can redistribute it
and/or modify it under the terms of the GNU General
Public License as published by the Free Software
Foundation; either version 2 of the License, or (at your
option) any later version.
```

```
\end{quote}
```

```
Bla bla bla bla bla...
```

```
\begin{quotation}
```

```

This program is distributed in the hope that it will be
useful, but WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A
```

```
PARTICULAR PURPOSE.  See the GNU General Public License
for more details.
```

```
You should have received a copy of the GNU General
Public License along with this program; if not, write
to the Free Software Foundation, Inc., 675 Mass Ave,
Cambridge, MA 02139, USA.
```

```
\end{quotation}
```

Bla bla bla bla bla...

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Bla bla bla bla bla...

This program is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

50.8.5 Versi

LaTeX facilita la scrittura di versi poetici attraverso l'ambiente '**verse**'. Si osservi l'esempio seguente:



50.8.6 Testo letterale



Per riprodurre del testo in modo letterale, è disponibile l'ambiente '**verbatim**' e il comando '**\verb**':

Comando	Descrizione
<pre>\begin{verbatim} <i>testo_preformattato</i> \end{verbatim}</pre>	<p>compone il testo in modo letterale, rispettando anche le interruzioni di riga, usando un carattere dattilografico;</p>
<pre>\begin{verbatim*} <i>testo_preformattato</i> \end{verbatim*}</pre>	<p>come nel caso precedente, evidenziando gli spazi con un segno speciale.</p>
<pre>\verb<i>xtesto_letterale</i>x</pre>	<p>compone il testo, che non può superare la riga di lunghezza, in modo letterale, usando un carattere dattilografico;</p>
<pre>\verb*<i>xtesto_letterale</i>x</pre>	<p>come nel caso precedente, evidenziando gli spazi con un segno speciale.</p>

Come si può comprendere dalla spiegazione data nella tabella riepilogativa, l'ambiente '**verbatim**' consente di rappresentare facilmente dei blocchi di testo letterali, da riprodurre così come sono nel sorgente, anche nelle interruzioni di riga; al contrario, il comando '**\verb**' va usato nel testo normale.

Nel caso dell'ambiente '**verbatim**', il contenuto del testo non ha limitazioni, salvo il fatto che non può includere il comando di chiusura dell'ambiente ('**\end{verbatim}**'); mentre il comando '**\verb**'

delimita il testo letterale con un carattere a scelta, inserito all'inizio e alla fine del testo. Gli esempi seguenti mostrano l'uso dei due modi di indicare testo letterale, in modo normale o con l'asterisco:

```
\$ \verb@cat -n /bin/bash@ [Invio]

\begin{verbatim}
 1  ^?ELF^A^A^A^@^@^@^@^@^@^@^@^@^B^@^C^@^A^@^@^@M-PM-^Q
 2  ^F^@^@M-w^B^@^@^_ ^F^@^@M-^^^D^@^@^@^@^@^@^F^@^@^K^
 3  ^D^@^@b^F^@^@^@^@^@^@^@^@_ ^C^@^@M-^[ ^C^@^@M-^[ ^@^@^@M-^J
  ...
\end{verbatim}
```

\$ cat -n /bin/bash [Invio]

```
 1  ^?ELF^A^A^A^@^@^@^@^@^@^@^@^@^B^@^C^@^A^@^@^@M-PM-^Q
 2  ^F^@^@M-w^B^@^@^_ ^F^@^@M-^^^D^@^@^@^@^@^@^F^@^@^K^
 3  ^D^@^@b^F^@^@^@^@^@^@^@^@_ ^C^@^@M-^[ ^C^@^@M-^[ ^@^@^@M-^J
  ...
```

```
\$ \verb*@ls -l /@ [Invio]
```

```
\begin{verbatim*}
drwxr-xr-x    2 root    root          4096 mag  3 22:46 bin
drwxr-xr-x    4 root    root          4096 apr 20 16:15 boot
drwxr-xr-x    7 root    root        20480 mag  5 08:49 dev
drwxr-xr-x  123 root    root          8192 mag  5 08:49 etc
drwxr-xr-x   21 root    root          4096 feb 10 20:59 home
drwxr-xr-x    7 root    root          8192 mag  3 22:48 lib
  ...
\end{verbatim*}
```

`$ ls -l / [Invio]`

```
drwxr-xr-x 2 root root 4096 mag 3 22:46 bin
drwxr-xr-x 4 root root 4096 apr 20 16:15 boot
drwxr-xr-x 7 root root 20480 mag 5 08:49 dev
drwxr-xr-x 123 root root 8192 mag 5 08:49 etc
drwxr-xr-x 21 root root 4096 feb 10 20:59 home
drwxr-xr-x 7 root root 8192 mag 3 22:48 lib
...
```

Si veda anche la sezione [50.14.2](#) per l'importazione di file esterni da rappresentare in modo letterale.

50.8.7 Ambienti fluttuanti

LaTeX dispone di due ambienti fluttuanti nel testo, preparati idealmente per incorporare tabelle o figure: «

```
\begin{figure} [permesso_di_collocazione ]
  figura
\end{figure}
```

```
\begin{table} [permesso_di_collocazione ]
  tabella
\end{table}
```

Per il momento, per come sono mostrati i modelli sintattici dell'utilizzo di questi ambienti, non c'è differenza tra un ambiente o l'altro. La collocazione dell'oggetto contenuto nell'ambiente fluttuante, viene concessa attraverso l'uso di alcune lettere alfabetiche, con il significato seguente:

Lettera	Significato mnemonico	Descrizione
h	<i>here</i>	si concede la collocazione nel punto in cui si trova la dichiarazione della tabella o della figura;
t	<i>top</i>	si concede la collocazione all'inizio della pagina attuale;
b	<i>bottom</i>	si concede la collocazione alla fine della pagina attuale;
p	<i>page</i>	si concede la collocazione in una pagina a parte, riservata a questo.

Queste lettere possono essere sommate assieme e conta l'ordine in cui sono dati i permessi di collocazione. Per esempio,

```
\begin{table} [tbp]
...
\end{table}
```

dichiara una tabella fluttuante che può essere collocata nell'ordine: all'inizio della pagina, alla fine, oppure in una pagina a parte. In pratica, viene esclusa la collocazione nel punto in cui viene dichiarata, a meno che questo coincida con una delle alternative concesse.

Se i permessi di posizionamento dell'oggetto non vengono specificati, LaTeX intende l'uso della sigla '**tbp**', cioè in alto, in basso o in una nuova pagina, escludendo la posizione naturale.

Dal momento che un oggetto inserito in un ambiente fluttuante non può essere suddiviso, risulta difficile imporre la collocazione nello stesso punto in cui si trova, quando ciò costringe a lasciare molto spazio vuoto alla fine di una pagina. Pertanto, l'uso del solo per-

messo ‘**h**’, senza alternative, diventa privo del suo significato nella maggior parte dei casi. Se la collocazione di questi oggetti deve avvenire necessariamente nella loro posizione naturale, lo si può imporre incorporando il pacchetto ‘**float**’ nel preambolo e indicando il permesso ‘**H**’ (la lettera H maiuscola):

```
\usepackage{float}
\begin{document}
...
\begin{table}[H]
...
\end{table}
```

La differenza tra gli ambienti si avverte solo quando si inseriscono le didascalie, attraverso il comando ‘**\caption**’:

```
\begin{figure}[permesso_di_collocazione]
  \caption{didascalia}
  figura
\end{figure}
```

```
\begin{table}[permesso_di_collocazione]
  \caption{didascalia}
  tabella
\end{table}
```

Il comando ‘**\caption**’ può essere inserito in qualunque punto, all’interno dell’ambiente, purché ciò possa avere senso in fase di composizione (generalmente si sceglie l’inizio o la fine, per avere una didascalia superiore o inferiore). L’inserimento del comando fa incrementare il contatore delle figure o delle tabelle, corrispondente

rispettivamente a **'figure'** o **'table'**; inoltre la didascalia risulta arricchita della definizione «figura» o «tabella» con l'aggiunta del numero relativo. Infine, il numero della figura o della tabella viene accumulato per generare un indice specifico, attraverso i comandi **'\listoffigures'** e **'\listoftables'**.

L'esempio seguente mostra la costruzione di una tabella molto povera, che serve a comprendere quanto si indifferente per LaTeX cosa sia il contenuto effettivo di questi ambienti:

```
\begin{table}[htbp]
  \begin{verbatim}
      .------.
      | n. | cognome | nome           | professione |
      |----|-----|-----|-----|
      |  1 | Tizi   | Tizio         | tramviere  |
      |----|-----|-----|-----|
      |  2 | Cai   | Caio         | pasticcere |
      |----|-----|-----|-----|
      |  3 | Semproni | Sempronio    | falegname  |
      `-----'
  \end{verbatim}
  \caption{Una tabella molto semplice}
\end{table}
```

```

.------.
| n. | cognome | nome | professione |
|----|-----|-----|-----|
| 1 | Tizi | Tizio | tramviere |
|----|-----|-----|-----|
| 2 | Cai | Caio | pasticcere |
|----|-----|-----|-----|
| 3 | Semproni | Sempronio | falegname |
‘-----’

```

Tabella 1: Una tabella molto semplice

Nella sezione [50.6.2](#) sono descritti i comandi con cui si può ottenere un salto pagina. In particolare, i comandi ‘`\clearpage`’ e ‘`\cleardoublepage`’ si occupano anche di inserire gli oggetti fluttuanti che non fossero già stati collocati secondo i permessi attribuiti. In pratica, se degli oggetti fluttuanti non hanno ancora potuto essere collocati, questi comandi per il salto pagina forzano il loro l’inserimento nella composizione, probabilmente in una pagina a parte. Di solito, i comandi che in qualche modo implicano un salto pagina, fanno uso di questi comandi ‘`\clearpage`’ o ‘`\cleardoublepage`’, proprio per chiudere anche con gli oggetti fluttuanti ancora sospesi.

50.9 Spazi e scatole

LaTeX evita di dare la stessa libertà che dà TeX nella gestione degli spazi e delle scatole. In questo senso, sono disponibili una serie di comandi, spesso simili tra loro, che possono creare un po’ di confusione all’utente. «

50.9.1 Spazi orizzontali



A proposito di spazi orizzontali, esistono ovviamente due possibilità: spazi rigidi e spazi elastici. Nel secondo caso, sono disponibili anche comandi che inseriscono puntini o linee in modo elastico, come quando si compila un indice e si vuole guidare la vista verso il numero della pagina.

Comando	Descrizione
<code>\hspace{<i>lunghezza</i>}</code>	Inserisce uno spazio orizzontale rigido. Il valore dell'argomento può essere negativo; in tal caso si ottiene un arretramento. Se lo spazio inserito viene a trovarsi all'inizio di una riga, viene ignorato, a meno che sia usato l'asterisco.
<code>\hspace*{<i>lunghezza</i>}</code>	

L'esempio seguente vuole mettere a confronto uno spazio normale, che viene soppresso perché si trova all'inizio di una riga, con uno spazio con asterisco, che risulta inserito in ogni caso:

```
1 Bla bla bla \hspace{3cm} bla bla bla...
```

```
2 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla \hspace{5cm} bla bla
bla...
```

```
3 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla \hspace*{5cm} bla bla
bla...
```

```
1 Bla bla bla                bla bla bla...
```

```
2 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla                bla bla bla...
```

```
3 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla                bla bla bla...
```


Comando	Descrizione
<code>\hfill</code>	Inserisce uno spazio orizzontale elastico che, in condizioni normali, va a riempire tutto lo spazio disponibile.
<code>\dotfill</code>	Inserisce uno spazio orizzontale elastico contenente dei puntini.
<code>\hrulefill</code>	Inserisce uno spazio orizzontale elastico contenente una linea orizzontale.

Generalmente, lo spazio orizzontale elastico si usa per allineare qualcosa al centro o a destra, come nell'esempio seguente:

```
\noindent sinistra\hfill centro\hfill destra\par
\noindent sinistra\dotfill centro\hrulefill destra\par
```

Come si vede dal risultato, questi comandi hanno tutti lo stesso grado di elasticità:

```
sinistra                centro                destra
sinistra.....centro_____destra
```

È importante osservare che i comandi per gli spazi elastici non funzionano se si trovano all'inizio o alla fine di una riga. Eventualmente, se è necessario, si può inserire uno spazio non interrompibile ('~') prima o dopo il comando:

```
\noindent ~\hrulefill destra\par
\noindent sinistra\dotfill ~\par
```

```
_____destra
sinistra.....
```

50.9.2 Spazi verticali

I comandi per l'inserimento di spazi verticali riprendono la stessa logica di quelli orizzontali (escludendo il riempimento con puntini o linee). In generale, gli spazi rigidi sono cumulabili.

Comando	Descrizione
<code>\vspace{<i>lunghezza</i>}</code>	Inserisce uno spazio verticale rigido. Se si trova all'inizio di una pagina, viene ignorato.
<code>\vspace*{<i>lunghezza</i>}</code>	Inserisce uno spazio verticale rigido. Se si trova all'inizio di una pagina, viene inserito regolarmente.
<code>\addvspace{<i>lunghezza</i>}</code>	Aggiunge uno spazio verticale rigido. Se nello stesso punto vengono usati più comandi ' <code>\addvspace</code> ', viene preso in considerazione solo quello con la lunghezza maggiore.
<code>\vfill</code>	Inserisce uno spazio verticale elastico, che in condizioni normali va a riempire tutto lo spazio disponibile.

L'esempio seguente riguarda l'inserzione di spazi rigidi attraverso i comandi '`\vspace`' e '`\addvspace`':

```

1 Bla bla bla...

\vspace{0.5cm}
2 Bla bla bla...

\vspace{0.5cm}
\vspace{0.5cm}
3 Bla bla bla...

\vspace{0.5cm}
\addvspace{0.5cm}
\addvspace{1.5cm}
4 Bla bla bla...

```

Si può osservare che i comandi '`\vspace`' si sommano tra loro,

mentre ‘`\addvspace`’ no, limitando ad aggiungere lo spazio più grande:

1 Bla bla bla...

2 Bla bla bla...

3 Bla bla bla...

4 Bla bla bla...

Lo spazio verticale elastico consente di allineare il testo verticalmente, come nell’esempio seguente, in cui si individua un blocco centrale e un blocco in fondo alla pagina:

```
1 Sopra, bla bla bla...
```

```
\vfill
```

```
2 Centro, bla bla bla...
```

```
\vfill
```

```
3 Sotto, bla bla bla...
```

```
\newpage
```

1 Sopra, bla bla bla...

2 Centro, bla bla bla...

3 Sotto, bla bla bla...

Per l'uniformità della veste grafica del testo, è importante standardizzare gli spazi verticali, attraverso dei comandi appositi, che cambiano a seconda dello stile generale:

Comando	Descrizione
<code>\smallskip</code>	spazio piccolo;
<code>\medskip</code>	spazio medio;
<code>\bigskip</code>	spazio grande.

L'entità dello spazi verticale introdotto da questi comandi dipende da altri comandi:

Comando	Descrizione
<code>\smallskipamount</code>	lunghezza piccola;
<code>\medskipamount</code>	lunghezza media;
<code>\bigskipamount</code>	lunghezza grande.

In pratica, i comandi per l'inserzione di spazi verticali standard corrispondono all'uso di '**\vskip**' con queste lunghezze uniformi:

Comando	Equivalenza	Esempio di altezza
<code>\smallskip</code>	<code>\vspace{\smallskipamount}</code>	3.0pt plus 1.0pt minus 1.0pt
<code>\medskip</code>	<code>\vspace{\medskipamount}</code>	6.0pt plus 2.0pt minus 2.0pt
<code>\bigskip</code>	<code>\vspace{\bigskipamount}</code>	12.0pt plus 4.0pt minus 4.0pt

L'esempio seguente mostra in pratica l'uso di questi comandi per l'inserzione di spazi verticali standard:

```
Bla bla bla...\bigskip

Lo spazio prima di questa riga è grande;\medskip

lo spazio prima di questa riga è medio;\smallskip

lo spazio prima di questa riga è piccolo;

lo spazio prima di questa riga è quello normale.
```

Bla bla bla...

Lo spazio prima di questa riga è grande;

lo spazio prima di questa riga è medio;

lo spazio prima di questa riga è piccolo;

lo spazio prima di questa riga è quello normale.

50.9.3 Elasticità controllabile

È possibile definire in modo semplice una lunghezza elastica da usare poi con comandi come '**\hspace**' e '**\vspace**':

Comando	Descrizione
<code>\stretch{n}</code>	restituisce una lunghezza elastica pari a <i>n</i> .

Maggiore è il valore attribuito all'argomento di '`\stretch`', maggiore è di conseguenza l'elasticità. A titolo di esempio si può vedere come si trasforma questo comando:

Comando	Lunghezza equivalente
<code>\stretch{1}</code>	0.0ptplus1fill
<code>\stretch{2}</code>	0.0ptplus2fill
<code>\stretch{2.5}</code>	0.0ptplus2.5fill

50.9.4 Interlinea



La distanza tra le righe di un blocco di testo normale si può controllare modificando la lunghezza '`\baselineskip`'; tuttavia, con l'aiuto del pacchetto '`setspace`', si possono usare comandi che semplificano questo tipo di intervento:

Comando	Descrizione
<code>\usepackage{setspace}</code>	dichiara l'utilizzo del pacchetto necessario alla gestione dei comandi sottostanti;
<code>\singlespacing</code>	richiede un'interlinea normale;
<code>\onehalfspacing</code>	richiede un'interlinea maggiore;
<code>\doublespacing</code>	richiede un'interlinea doppia.

L'esempio seguente mostra l'uso e l'effetto di questi comandi:

```

...
\usepackage{setspace}
...
\begin{document}

\singlespacing
Questo paragrafo ha un'interlinea normale. Bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla...

\onehalfspacing
Questo paragrafo ha un'interlinea pari a una volta e mezza
quella normale. Bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

\doublespacing
Questo paragrafo ha un'interlinea doppia. Bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla...

\end{document}

```

Questo paragrafo ha un'interlinea normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Questo paragrafo ha un'interlinea pari a una volta e mezza quella normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Questo paragrafo ha un'interlinea doppia. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Anche senza il pacchetto '**setspace**', si può usare nel preambolo il comando seguente:

Comando	Descrizione
<code>\linespread{<i>fattore</i>}</code>	moltiplica la distanza tra le righe del fattore indicato.

Per esempio, così:

```
...
\linespread{2.5}
...
\begin{document}

La distanza tra le righe è di due volte e mezza quella
normale. Bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla...

\end{document}
```

La distanza tra le righe è di due volte e mezza quella normale. Bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla...

50.9.5 Scatole



LaTeX offre una serie di comandi per la realizzazione di scatole, bordate e non, alcuni dei quali sono già apparsi in altri capitoli. Oltre alla presenza del bordo, le scatole si distinguono per la capacità o meno di rimpaginano il testo al loro interno.

Comando	Comando	Descrizione
<code>\mbox{<i>testo</i>}</code>	<code>\fbox{<i>testo</i>}</code>	Inserisce il testo in una scatola larga quanto il testo stesso; nel secondo caso si ottiene una scatola bordata.
<code>\makebox[<i>ampiezza</i>]{<i>testo</i>}</code> <code>\makebox[<i>ampiezza</i>]{<i>testo</i>}</code>	<code>\framebox[<i>ampiezza</i>]{<i>testo</i>}</code> <code>\framebox[<i>ampiezza</i>]{<i>testo</i>}</code>	Inserisce il testo (di una sola riga) in una scatola; nel secondo caso si ottiene una scatola bordata.

Quelli appena mostrati sono i comandi più comuni per ottenere delle scatole, in cui il testo occupa sempre una riga. Si può intuire che i comandi ‘`\makebox`’ e ‘`\framebox`’, usati senza argomenti opzionali, sono equivalenti rispettivamente a ‘`\mbox`’ e ‘`\fbox`’. Il secondo argomento opzionale di ‘`\makebox`’ e ‘`\framebox`’ è una lettera che indica l’allineamento orizzontale che deve avere il testo contenuto:

Lettera	Significato mnemonico	Descrizione
c	<i>center</i>	testo centrato orizzontalmente nella scatola;
l	<i>left</i>	testo allineato a sinistra;
r	<i>right</i>	testo allineato a destra;
s	<i>stretch</i>	testo allargato per tutta l’ampiezza della scatola.

L’esempio seguente dovrebbe chiarire l’utilizzo di questi comandi:

```
Bla bla bla \mbox{ciao ciao} bla bla bla \fbox{ciao ciao}
bla bla bla...
```

```
Bla bla bla \makebox{ciao ciao} bla bla bla \framebox{ciao
ciao} bla bla bla...
```

```
Bla bla bla \makebox[4cm]{ciao ciao} bla bla bla
\framebox[4cm]{ciao ciao} bla bla bla...
```

```
Bla bla bla \makebox[4cm][c]{ciao ciao} bla bla bla
\framebox[4cm][c]{ciao ciao} bla bla bla...
```

```
Bla bla bla \makebox[4cm][l]{ciao ciao} bla bla bla
\framebox[4cm][l]{ciao ciao} bla bla bla...
```

```
Bla bla bla \makebox[4cm][r]{ciao ciao} bla bla bla
\framebox[4cm][r]{ciao ciao} bla bla bla...
```

```
Bla bla bla \makebox[4cm][s]{ciao ciao} bla bla bla
\framebox[4cm][s]{ciao ciao} bla bla bla...
```

Bla bla bla ciao ciao bla bla bla ciao ciao bla bla bla...

Bla bla bla ciao ciao bla bla bla ciao ciao bla bla bla...

Bla bla bla ciao ciao bla bla bla ciao ciao
bla bla bla...

Bla bla bla ciao ciao bla bla bla ciao ciao
bla bla bla...

Bla bla bla ciao ciao bla bla bla ciao ciao
bla bla bla...

Bla bla bla ciao ciao bla bla bla ciao ciao
bla bla bla...

Bla bla bla ciao ciao bla bla bla ciao ciao
bla bla bla...

Le scatole bordate con ‘**\fbox**’ e ‘**\framebox**’ sono controllate da

due dimensioni, molto utili per gestirne l'estetica:

Comando	Descrizione
$\backslash\text{fboxrule}=\textit{dimensione}$ $\backslash\text{setlength}\{\backslash\text{fboxrule}\}\{\textit{dimensione}\}$	definisce lo spessore della linea usata per il bordo;
$\backslash\text{fboxsep}=\textit{dimensione}$ $\backslash\text{setlength}\{\backslash\text{fboxsep}\}\{\textit{dimensione}\}$	definisce la distanza tra la linea del bordo e il contenuto della scatola.

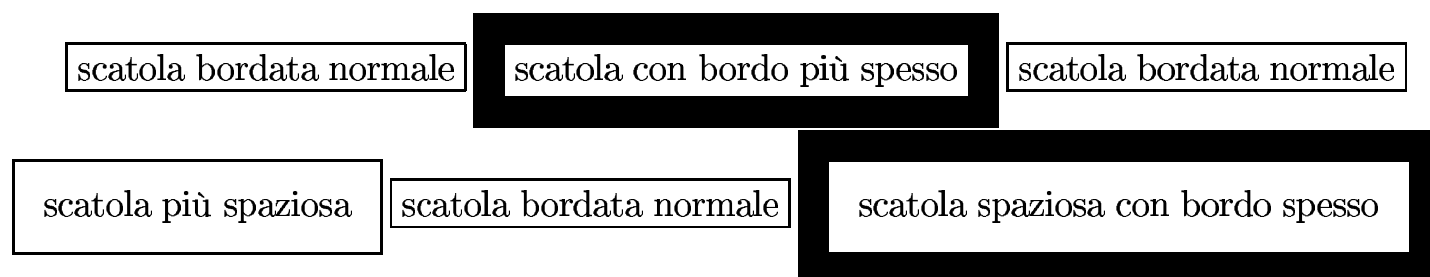
L'esempio seguente dovrebbe chiarire in che modo usare questi valori per controllare le scatole bordate:

```

\fbx{scatola bordata normale}
{\fboxrule=3mm\fbx{scatola con bordo più spesso}}
\fbx{scatola bordata normale}
{\fboxsep=3mm\fbx{scatola più spaziosa}}
\fbx{scatola bordata normale}
{\fboxrule=3mm\fboxsep=3mm\fbx{scatola spaziosa con bordo
spesso}}

```

Si può osservare che le scatole «speciali» sono state racchiuse tra parentesi graffe, per evitare che la modifica delle caratteristiche legate allo spessore della linea e alla distanza dal contenuto, possa coinvolgere il resto del testo. L'alternanza con scatole normali dimostra il funzionamento di questo sistema di contenimento:



Per inserire in una scatola più di una riga di testo, si possono usare

due tipi di istruzioni, ottenendo risultati leggermente differenti:

<pre>\parbox[c t b][<i>altezza</i>]\leftarrow \rightarrow[c t b s]{<i>larghezza</i>}{<i>testo</i>}</pre>	<p>definisce una scatola, senza bordo, per contenere del testo impaginato nell'ambito della larghezza stabilita;</p>
<pre>\begin{minipage}[c t b]{<i>larghezza</i>} <i>testo</i> \end{minipage}</pre>	<p>definisce una scatola, senza bordo, per contenere del testo impaginato nell'ambito della larghezza stabilita.</p>

La differenza più importante tra questi due tipi di scatole sta nel fatto che ‘**\parbox**’ consente l’inserimento di testo non troppo complesso, mentre l’ambiente ‘**minipage**’ è letteralmente una piccola pagina, in cui anche le note a piè pagina rimangono al suo interno.

In entrambi i casi, il primo argomento opzionale indica la collocazione della scatola, in senso verticale, rispetto alla base della riga in cui ci si trova; pertanto:

Lettera	Significato mnemonico	Descrizione
c	<i>center</i>	richiede una scatola centrata verticalmente rispetto alla riga;
t	<i>top</i>	richiede una scatola che parte dalla riga e si espande in basso;
b	<i>bottom</i>	richiede una scatola che parte dalla riga e si espande in alto.

L’esempio seguente consente di confrontare il comportamento dei due tipi di scatole per ciò che riguarda l’allineamento di queste rispetto alla riga. Per facilitare il paragone visivo, queste scatole

vengono bordate attraverso il comando '**\fbox**':

```
Bla bla bla bla bla...
```

```
Bla bla \fbox{\parbox{2cm}{Questa frase è in una scatola}}  
bla \fbox{\parbox[c]{2cm}{Questa frase è in una scatola}}  
bla \fbox{\parbox[t]{2cm}{Questa frase è in una scatola}}  
bla \fbox{\parbox[b]{2cm}{Questa frase è in una scatola}}  
bla bla.
```

```
Bla bla bla bla bla...
```

```
Bla bla \fbox{\begin{minipage}{2cm}Questa frase è in una  
scatola\end{minipage}} bla  
\fbox{\begin{minipage}[c]{2cm}Questa frase è in una  
scatola\end{minipage}} bla  
\fbox{\begin{minipage}[t]{2cm}Questa frase è in una  
scatola\end{minipage}} bla  
\fbox{\begin{minipage}[b]{2cm}Questa frase è in una  
scatola\end{minipage}} bla bla.
```

```
Bla bla bla bla bla...
```

Come si vede dal risultato della composizione, per il momento non si notano differenze:

Bla bla bla bla bla...

Bla bla Questa frase è in una scatola bla Questa frase è in una scatola bla Questa frase è in una scatola bla Questa frase è in una scatola

bla bla.

Bla bla bla bla bla...

Bla bla Questa frase è in una scatola bla Questa frase è in una scatola bla Questa frase è in una scatola bla Questa frase è in una scatola

bla bla.

Bla bla bla bla bla...

Dalla sintassi mostrata a proposito di ‘`\parbox`’, si nota la possibilità di indicare l’altezza della scatola, assieme a un terzo argomento opzionale che serve a collocare il testo nella scatola:

Lettera	Significato mnemonico	Descrizione
c	<i>center</i>	testo centrato verticalmente nella scatola;
t	<i>top</i>	testo in alto;
b	<i>bottom</i>	testo in basso;
s	<i>stretch</i>	testo allungato per tutta l’altezza della scatola.

L’esempio seguente riguarda esclusivamente ‘`\parbox`’, per mostrare l’uso dei due ulteriori argomenti opzionali:

```
Bla bla bla bla bla...
```

```
Bla bla bla \fbox{\parbox[c][2cm]{2cm}{Questa frase è in una scatola}}
```

```

\fbbox{\parbox[c][2cm][c]{2cm}{Questa frase è in una scatola}}
\fbbox{\parbox[c][2cm][t]{2cm}{Questa frase è in una scatola}}
\fbbox{\parbox[c][2cm][b]{2cm}{Questa frase è in una scatola}}
\fbbox{\parbox[c][2cm][s]{2cm}{Questa frase è in una scatola}}

Bla bla bla bla bla...

```

Si può osservare che nell'ultima scatola, l'opzione di allungamento verticale del testo non funziona, perché il testo non possiede spazi elastici:

Bla bla bla bla bla...

Bla bla bla

Questa frase
è in una sca-
tola

Questa frase
è in una sca-
tola

Questa frase
è in una sca-
tola

Questa frase
è in una sca-
tola

Questa frase
è in una sca-
tola

Bla bla bla bla bla...

50.9.6 Testo da salvare e da recuperare

LaTeX ha una capacità limitata di memorizzare del testo in modo da poterlo inserire successivamente nella composizione:

Comando	Descrizione
<code>\begin{lrbox}{<i>n</i>}testo\end{lrbox}</code>	accumula nel registro <i>n</i> il testo indicato;
<code>\usebox{<i>n</i>}</code>	inserisce nella composizione il testo corrispondente al registro <i>n</i> , in una scatola orizzontale.

La spiegazione dei due modelli sintattici dovrebbe essere sufficiente a comprendere il funzionamento della cosa; tuttavia è importante sottolineare che il testo accumulato può essere composto da una sola riga e viene poi rappresentato come se fosse inserito in una scatola ‘**\mbox**’. Segue un esempio:

```
\begin{lrbox}{1}
ciao a tutti
\end{lrbox}

\begin{lrbox}{2}
ma che bello
\end{lrbox}

Bla bla bla \usebox{2} bla bla bla \usebox{1} bla bla bla
\usebox{2} bla bla bla \usebox{1} bla bla bla...
```

Bla bla bla ma che bello bla bla bla ciao a tutti bla bla bla ma che bello bla bla bla ciao a tutti bla bla bla...

50.9.7 Linee

«

Con LaTeX, si ottengono linee orizzontali o verticali con il comando ‘**\rule**’. Per la precisione, si disegnano dei rettangoli neri di ampiezza e altezza stabiliti; pertanto, per ottenere una linea orizzontale si utilizza un rettangolo molto largo e molto basso, mentre per una linea verticale si usa un rettangolo molto stretto. Questi rettangoli sono collocati a loro volta nella parte superiore di una scatola, che in condizioni normali è alta quanto basta per contenere l’oggetto.

```
\rule{larghezza}{altezza}
```



```
\rule[spazio_verticale_aggiuntivo] { larghezza } { altezza }
```

L'esempio seguente riguarda la realizzazione di alcune linee, verticali e orizzontali, senza spazi verticali aggiuntivi:

```
Bla bla bla...
```

```
Bla bla bla \rule{0.5mm}{1cm} \rule{0.6mm}{1cm}
\rule{0.7mm}{1cm} \rule{0.8mm}{1cm} \rule{0.9mm}{1cm}
\rule{1mm}{1cm} bla bla bla...
```

```
\rule{\textwidth}{0.5mm}
```

```
\rule{\textwidth}{0.6mm}
```

```
\rule{\textwidth}{0.7mm}
```

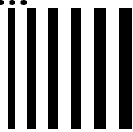
```
\rule{\textwidth}{0.8mm}
```

```
\rule{\textwidth}{0.9mm}
```

```
\rule{\textwidth}{1mm}
```

```
Bla bla bla...
```

Bla bla bla...



Bla bla bla bla bla bla...

Bla bla bla...

L'esempio seguente è una variante con l'aggiunta di spazi verticali aggiuntivi:

```
Bla bla bla...
```

```
Bla bla bla \rule[1mm]{0.5mm}{1cm} \rule[2mm]{0.6mm}{1cm}
\rule[3mm]{0.7mm}{1cm} \rule[4mm]{0.8mm}{1cm}
\rule[5mm]{0.9mm}{1cm} \rule[6mm]{1mm}{1cm} bla bla bla...
```

```
\rule[1mm]{\textwidth}{0.5mm}
```

```
\rule[2mm]{\textwidth}{0.6mm}
```

```
\rule[3mm]{\textwidth}{0.7mm}
```

```
\rule[4mm]{\textwidth}{0.8mm}
```

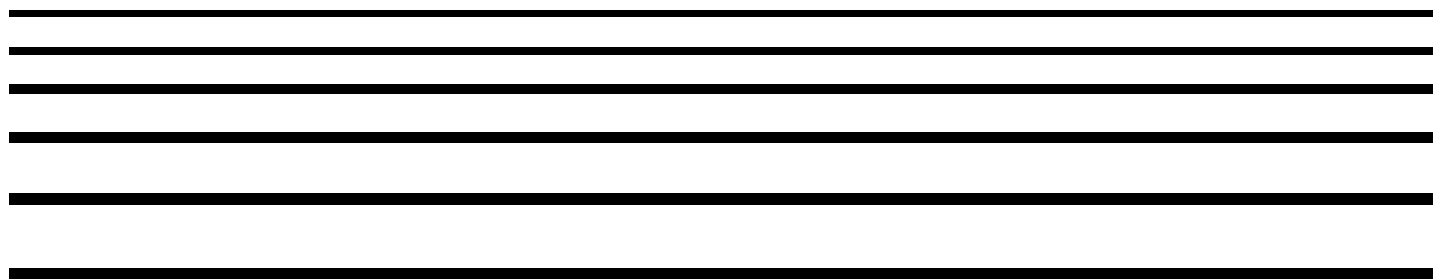
```
\rule[5mm]{\textwidth}{0.9mm}
```

```
\rule[6mm]{\textwidth}{1mm}
```

```
Bla bla bla...
```

Bla bla bla...

Bla bla bla  bla bla bla...



Bla bla bla...

50.10 Riferimenti incrociati

LaTeX gestisce diversi tipi di riferimenti incrociati, di cui il più importante è l'indice generale, ottenuto attraverso il comando `\tableofcontents`. La gestione dei riferimenti incrociati avviene attraverso la creazione di file temporanei, adibiti ad accumulare le informazioni, da riutilizzare attraverso rielaborazioni successive. In pratica, è proprio la presenza di questi riferimenti incrociati che obbliga a ripetere il procedimento di composizione più volte.

50.10.1 Indice generale, indice delle tabelle e delle figure

La creazione dell'indice generale, dell'indice delle tabelle e dell'indice delle figure, parte dall'accumulo delle informazioni sulla collocazione di queste voci. Per la precisione, i comandi che definiscono la suddivisione del documento annotano il titolo, il numero della sezione e la pagina in un file con estensione `.toc` (*Table of contents*); i comandi `\caption` inseriti nell'involucro fluttuante di una tabel-

la o di una figura, annotano il numero e la pagina in cui si trovano in file con estensione ‘.lot’ (*List of tables*) e ‘.lof’ (*List of figures*).

Nella rielaborazione successiva (quando il comando di composizione viene ripetuto), questi file possono essere incorporati attraverso i comandi ‘`\tableofcontents`’, ‘`\listoftables`’ e ‘`\listoffigures`’, per ottenere rispettivamente l’indice generale, l’indice delle tabelle e delle figure.

Dal momento che l’inserzione modifica la lunghezza del testo, a meno che gli indici siano collocati alla fine del documento, si rende necessaria almeno una terza rielaborazione.

Comando	Descrizione
<code>\tableofcontents</code>	inserisce nella composizione l’indice generale, ottenuto dal file ‘.toc’, generato dalla composizione precedente;
<code>\listoftables</code>	inserisce nella composizione l’indice delle tabelle, ottenuto dal file ‘.lot’, generato dalla composizione precedente;
<code>\listoffigures</code>	inserisce nella composizione l’indice delle figure, ottenuto dal file ‘.lof’, generato dalla composizione precedente.

L’inserimento di voci in questi elenchi può avvenire anche in modo manuale, attraverso due comandi appositi:

Comando	Descrizione
<code>\addtocontents{toc lot lof}{<i>testo</i>}</code>	aggiunge il testo indicato nell'ultimo argomento, nell'indice corrispondente alla sigla del primo argomento (indice generale, indice delle tabelle o indice delle figure);
<code>\addcontentsline↵ ↞{toc lot lof}{<i>livello</i>}{<i>testo</i>}</code>	aggiunge la voce nell'indice corrispondente alla sigla del primo argomento.

Il funzionamento del comando ‘`\addtocontents`’ è relativamente semplice, nel senso che ci si limita ad aggiungere del testo nell'indice prescelto:

```
\tableofcontents

\section{Prima}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

\addtocontents{toc}{Ciao a tutti\par}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

\section{Seconda}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...
```

Si può osservare che nel testo inserito è stata aggiunta la richiesta esplicita di concludere il paragrafo; ciò si rende necessario per evitare problemi imprevisti nel momento dell'incorporazione del file `‘.toc’`.

Indice

1	Prima	1
	Ciao a tutti	
2	Seconda	1

1 Prima

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

2 Seconda

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

L'uso di `‘\addcontentsline’` è più complesso, perché richiede anche l'indicazione del «livello» della voce. Se si tratta dell'indice generale, il livello è rappresentato dalle parole chiave da `‘part’` a `‘subparagraph’`, mentre negli altri casi è ammesso l'uso di `‘table’` o `‘figure’`. Così si intende che la voce inserita sia da trattare allo stesso modo del livello corrispondente, per ciò che riguarda la composizione. Nell'esempio che segue, la voce si inserisce al livello di una sottosezione:

```
\tableofcontents
```

```
\section{Prima}
```

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla  
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...
```

```
\addcontentsline{toc}{subsection}{Ciao a tutti}
```

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla  
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...
```

```
\section{Seconda}
```

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla  
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...
```

Come si vede, in questo caso non c'è più bisogno di concludere il paragrafo nel testo, cosa che al contrario genererebbe qui solo un errore:

50.10.2 Riferimenti liberi

«

LaTeX mette a disposizione pochi comandi per la creazione di riferimenti incrociati all'interno del testo. Attraverso il comando `'\label'` è possibile definire un'etichetta alla quale si può fare riferimento con i comandi `'\ref'` o `'\pageref'`:

Comando	Descrizione
<code>\label{stringa_identificativa}</code>	mette un punto di riferimento invisibile, identificato dalla stringa contenuta nell'argomento;
<code>\ref{stringa_identificativa}</code>	inserisce nella composizione il numero corrispondente alla stringa di identificazione indicata, secondo il contesto;
<code>\pageref{stringa_identificativa}</code>	inserisce nella composizione il numero della pagina corrispondente alla stringa di identificazione indicata.

La posizione in cui viene collocato il comando `'\label'` è importante, in quanto `'\ref'` si comporta diversamente a seconda del contesto. Se l'etichetta viene dichiarata all'interno di testo normale, il riferimento generico a questa restituisce un numero, più o meno articolato, che indica la sezione o il capitolo in cui si trova; se invece l'etichetta viene dichiarata all'interno di un ambiente numerato, come una tabella, una figura, un'equazione o altro, il riferimento a questa genera il numero corrispondente a tale elemento.

A titolo di esempio, viene mostrato come si potrebbe procedere per la costruzione manuale di un indice generale, senza l'ausilio del comando `'\tableofcontents'`:

```

\section*{Indice}

\noindent\ref{sec:Prima} Prima, pag. \pageref{sec:Prima}

\noindent\ref{sec:Seconda} Seconda, pag.
\pageref{sec:Seconda}

\section{Prima} \label{sec:Prima}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

\section{Seconda} \label{sec:Seconda}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

```

Indice

- 1 Prima, pag. 1
- 2 Seconda, pag. 1

1 Prima

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla...

2 Seconda

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla...

È bene ribadire che LaTeX inserisce implicitamente le etichette generate con il comando ‘`\label`’ in un contesto particolare, per cui

il risultato del comando ‘`\ref`’ cambia di conseguenza. Tuttavia, il numero che viene mostrato da ‘`\ref`’ non è accompagnato dal contesto in fase di composizione, pertanto è compito dell’autore scrivere di cosa si tratta (capitolo, sezione, tabella, figura o altro). In questo senso, diventa importante ricordare a cosa si riferisce una certa stringa di identificazione, pertanto, la stessa documentazione originale di LaTeX propone di usare un prefisso diverso in base al contesto:

Comando	Descrizione
<code>\label{cha:<i>stringa_successiva</i>}</code>	etichetta di identificazione di un capitolo;
<code>\label{sec:<i>stringa_successiva</i>}</code>	etichetta di identificazione di una sezione di livello inferiore al capitolo;
<code>\label{tab:<i>stringa_successiva</i>}</code>	etichetta di identificazione di una tabella;
<code>\label{fig:<i>stringa_successiva</i>}</code>	etichetta di identificazione di una figura;
<code>\label{eq:<i>stringa_successiva</i>}</code>	etichetta di identificazione di un’equazione.

Le stringhe di identificazione generate dal comando ‘`\label`’, sono annotate in un file con estensione ‘`.aux`’ (assieme ad altre informazioni estranee). Quando LaTeX viene riavviato in una fase di composizione successiva, carica in memoria i dati contenuti nel file ‘`.aux`’ e lo utilizza per risolvere i riferimenti con i comandi ‘`\ref`’ e ‘`\pageref`’.

50.10.3 Indice analitico



La realizzazione di un indice analitico, ovvero un indice che raccoglie una serie ordinata di voci che possono essere inserite in varie parti del documento, è più complessa rispetto a quella dell'indice generale. In special modo, in questo caso è necessario provvedere a riordinare e riorganizzare i dati nel modo corretto. I comandi coinvolti per la realizzazione dell'indice analitico, sono riassunti brevemente nell'elenco seguente:

Comando	Descrizione
<code>\usepackage{makeidx}</code>	si usa questo comando nel preambolo per importare il codice necessario alla gestione dell'indice analitico;
<code>\makeindex</code>	si usa questo comando, sempre nel preambolo, per attivare la gestione dell'indice analitico;
<code>\index{voce}</code>	inserisce la voce indicata nel file <code>.idx</code> , dal quale si ottiene poi l'indice analitico;
<code>\printindex</code>	inserisce l'indice analitico contenuto nel file <code>.ind</code> , ottenuto attraverso un'elaborazione esterna a LaTeX.

Come si vede, è necessario utilizzare un pacchetto esterno, denominato `makeidx`, quindi si deve attivare la costruzione dell'indice, con il comando `\makeindex`. Pertanto, si inseriscono le istruzioni seguenti nel preambolo:

```
\usepackage{makeidx}
\makeindex
```

Lo scopo particolare di `\makeindex` è quello di aprire un file con estensione `.idx`, in cui il comando `\index` può poi annotare le

Dopo l'elaborazione normale con LaTeX si ottiene un file con estensione `.idx`, con il contenuto seguente:

```
\indexentry{resistenza}{1}
\indexentry{induttanza}{1}
\indexentry{capacit\`a}{1}
\indexentry{resistenza}{1}
```

Si può osservare che le voci sono state inserite nella sequenza in cui appaiono nel sorgente LaTeX, senza tenere conto di eventuali ripetizioni. Ma LaTeX non è nelle condizioni di usare questo file direttamente nell'elaborazione successiva; prima occorre riordinare e riorganizzare le voci attraverso il programma `makeindex`:

```
makeindex nome_file_senza_estensione
```

Per esempio, se si sta facendo la composizione del file `prova.tex`, è sufficiente il comando:

```
$ makeindex prova [Invio]
```

In questo modo, `makeindex` cerca di leggere il file `prova.idx` (quello generato dall'elaborazione di LaTeX) e produce il file `prova.ind`, che, seguendo l'esempio, dovrebbe contenere quanto segue:

```
\begin{theindex}

  \item capacit\`a, 1

  \indexspace

  \item induttanza, 1

  \indexspace
```

```
\item resistenza, 1

\end{theindex}
```

Come si può vedere, la voce ‘**resistenza**’ contiene un riferimento unico, dal momento che appare due volte, ma nella stessa pagina finale.³

Se il sorgente LaTeX contiene il comando ‘**\printindex**’, in quel punto viene inserito il contenuto del file con estensione ‘.ind’, a partire da una pagina nuova, con l’aggiunta di un titolo adeguato:

```
...
\usepackage{makeidx}
\makeindex
...
\begin{document}

\section{Prima} \index{resistenza}\index{induttanza}

Qui si parla di resistenze e di induttanze, bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla...

\section{Seconda} \index{capacità}

Qui si parla di capacità e simili, bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla...

\printindex

\end{document}
```


1 Prima

Qui si parla di resistenze e di induttanze, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

2 Seconda

Qui si parla di capacità e ancora di resistenza, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Indice analitico

capacità, 1

induttanza, 1

resistenza, 1

1

È da osservare che la voce indicata come argomento del comando ‘`\index`’ può essere più articolata, in base alle capacità elaborative di ‘`makeindex`’. In pratica, si utilizzano dei simboli per separare la voce in più parti. L’elenco seguente dovrebbe aiutare a comprendere il meccanismo:

Comando	Descrizione
<code>\index{voce}</code>	questa è la situazione normale, in cui si inserisce la voce indicata, così come inserita, nell’indice;

Comando	Descrizione
<pre>\index{voce_principale!voce_secondaria}</pre>	<p>il punto esclamativo permette di separare una voce principale da una parte secondaria, che poi può essere mostrata con un leggero rientro rispetto alla prima parte;</p>
<pre>\index{voce_principale@voce_abbellita}</pre>	<p>la chiocciola separa una prima parte, da prendere in considerazione ai fini dell'ordinamento, dalla seconda parte che invece viene mostrata nell'indice, consentendo l'uso di comandi che modificano l'aspetto del carattere;</p>
<pre>\index{voce↔ ↔ comandi_per_il_numero_della_pagina}</pre>	<p>la barra verticale separa la voce dell'indice dai comandi per controllare l'aspetto del numero della pagina (comandi da usare senza la barra obliqua inversa iniziale).</p>

L'esempio seguente è una variante di quanto già visto, per mostrare l'uso del comando '**\index**' con questi simboli particolari:

```
\section{Prima}
\index{resistenza}\index{resistenza!induttiva|textit}
```

```
Qui si parla di resistenze e di induttanze, bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla
```

```
bla bla bla bla bla bla bla bla...
```

```
\section{Seconda} \index{resistenza!capacitiva|textit}
```

```
\index{resistenza@\texttt{resistenza}}
```

```
Qui si parla di capacità e ancora di resistenza, bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...
```

Ecco come si presenta l'indice; si può osservare che la voce **'resistenza'** appare due volte, perché il tipo di carattere è stato modificato:

Indice analitico

```
resistenza, 1
    capacitiva, 1
    induttiva, 1
resistenza, 1
```

1

50.10.4 Bibliografie

LaTeX offre una gestione molto semplice delle bibliografie attraverso l'uso dell'ambiente **'thebibliography'**. I comandi coinvolti sono descritti brevemente nell'elenco seguente:

Comando	Descrizione
<pre>\begin{thebibliography}↵ ↵{stringa_di_esempio}...↵ ↵\end{thebibliography}</pre>	<p>dichiara l'ambiente che contiene le informazioni bibliografiche;</p>

Comando	Descrizione
<code>\bibitem[<i>sigla</i>]↔</code> <code>↔{<i>stringa_identificativa_della_voce</i>}</code>	all'interno dell'ambiente 'thebibliography' dichiara l'inizio di una voce bibliografica;
<code>\cite[<i>testo_aggiuntivo</i>]↔</code> <code>↔{<i>stringa_identificativa_della_voce</i>}</code>	si usa al di fuori dell'ambiente 'thebibliography' , per richiamare il riferimento a una voce bibliografica.

In pratica, l'ambiente **'thebibliography'** serve a raccogliere le voci bibliografiche, ognuna dichiarata attraverso il comando **'\bibitem'**. Nel risultato della composizione, le voci vengono precedute da una sigla, come indicato dal comando **'\bibitem'**, oppure da un numero gestito in modo automatico; pertanto, per garantire un allineamento corretto, la dichiarazione dell'ambiente richiede un argomento composto da una stringa di esempio, che rappresenti l'ampiezza massima di questa sigla, sia gestita manualmente, sia automatica.

Al di fuori dell'ambiente **'thebibliography'**, di solito prima di questo, si usa il comando **'\cite'** per fare riferimento a una voce bibliografica, attraverso la stringa identificativa che usa lo stesso comando **'\bibitem'**. Se viene usato l'argomento opzionale di **'\cite'**, si ottiene l'aggiunta di quella stringa dopo la sigla usata per identificarlo.

Per cominciare conviene mostrare un esempio molto semplice, in cui non si usano argomenti opzionali:

```

\section{Resistenze e condensatori}

Qui si parla di resistenze \cite{res} e di condensatori
\cite{con}, bla bla bla bla bla bla bla bla...

\begin{thebibliography}{99}
\bibitem{res} Tizio Tizi, \textit{Resistenze elettriche},
2002
\bibitem{con} Caio Cai, \textit{Condensatori e correnti
alternate}, 2001
\end{thebibliography}

```

Come si vede, l'argomento della dichiarazione dell'ambiente **'thebibliography'** contiene la stringa **'99'**, perché si sa che la sigla usata per distinguere le voci non può essere più lunga di un numero di due cifre:

1 Resistenze e condensatori

Qui si parla di resistenze [1] e di condensatori [2], bla bla bla bla bla bla bla bla...

Riferimenti bibliografici

[1] Tizio Tizi, *Resistenze elettriche*, 2002

[2] Caio Cai, *Condensatori e correnti alternate*, 2001

Volendo aggiungere un riferimento a una certa pagina del primo libro citato, basta sfruttare l'argomento opzionale di **'\cite'**:

```

Qui si parla di resistenze \cite[pag. 1]{res} e di
condensatori \cite{con}, bla bla bla bla bla bla bla
bla...

```

1 Resistenze e condensatori

Qui si parla di resistenze [1, pag. 1] e di condensatori [2], bla bla bla bla bla bla bla bla bla...

Riferimenti bibliografici

[1] Tizio Tizi, *Resistenze elettriche*, 2002

[2] Caio Cai, *Condensatori e correnti alternate*, 2001

Infine, per usare delle sigle più significative, si può sfruttare l'argomento opzionale del comando '`\bibitem`', ma in questo caso occorre intervenire anche nella dichiarazione dell'ampiezza massima di queste stringhe:

```
\begin{thebibliography}{ZZZZZ}
\bibitem[TIZ]{res} Tizio Tizi, \textit{Resistenze
elettriche}, 2002
\bibitem[COND]{con} Caio Cai, \textit{Condensatori e
correnti alternate}, 2001
\end{thebibliography}
```

1 Resistenze e condensatori

Qui si parla di resistenze [TIZ, pag. 1] e di condensatori [COND], bla bla bla bla bla bla bla bla bla...

Riferimenti bibliografici

[TIZ] Tizio Tizi, *Resistenze elettriche*, 2002

[COND] Caio Cai, *Condensatori e correnti alternate*, 2001

Come si intuisce, l'uso di riferimenti bibliografici di questo tipo richiede delle rielaborazioni successive, anche se in questo caso non

intervengono programmi esterni.

Eventualmente, se questo tipo di gestione bibliografica si mostra insufficiente, è disponibile BibTeX, che consente di gestire una bibliografia molto dettagliata in un file esterno, dove solo le voci richieste espressamente vengono poi mostrate nella composizione finale.

50.11 Tabelle

LaTeX offre due ambienti per la costruzione di tabelle: ‘**tabbing**’ e ‘**tabular**’. Il primo dei due riguarda la realizzazione di tabelle attraverso delle tabulazioni, mentre il secondo è invece più completo, anche se poi offre meno libertà.

50.11.1 Tabella fatta di tabulazioni

Attraverso l’ambiente ‘**tabbing**’, si può allineare il testo in colonne, fissando gli stop di tabulazione attraverso scatole orizzontali di riferimento, che di solito si ottengono con del testo opportuno. Si osservi subito un esempio molto semplice:

```
\begin{tabbing}
marrone \= 10 \% \= quattro \kill

nero \> 0 \> zero \\
marrone \> 1 \> uno \\
rosso \> 2 \> due \\
arancio \> 3 \> tre \\
giallo \> 4 \> quattro \\
verde \> 5 \> cinque \\
blu \> 6 \> sei \\
viola \> 7 \> sette \\
grigio \> 8 \> otto \\
bianco \> 9 \> nove
```

```

argento \> 10 \% \\
oro \> 5 \% \\
\> 2 \% \\
\> 1 \% \\
\end{tabbing}

```

Si può già comprendere intuitivamente il significato delle istruzioni inserite nell'esempio. In particolare, si può osservare nel risultato della composizione che la prima riga non appare, ma serve solo per indicare alle righe successive la collocazione degli stop di tabulazione; infatti, si può vedere che la prima riga contiene l'esempio del testo più lungo utilizzato in ogni colonna:

nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre
giallo	4	quattro
verde	5	cinque
blu	6	sei
viola	7	sette
grigio	8	otto
bianco	9	nove
argento	10 %	
oro	5 %	
	2 %	
	1 %	

La sintassi per l'utilizzo di questo ambiente **'tabbing'** è rappresentata in modo semplificato dal modello seguente:


```

\begin{tabbing}
testo \= testo [\= testo] ... \kill
testo \> testo [\> testo] ... \\
[testo \> testo [\> testo] ... \\]
...
\end{tabbing}

```

Il modello non esaurisce tutte le possibilità, ma rende l'idea dell'utilizzo più frequente: all'inizio si dichiara la distanza degli stop di tabulazione con una riga di esempio, che poi non viene prodotta nella composizione. Tra le alternative possibili, c'è da considerare la possibilità di definire gli stop di tabulazione in una riga che poi viene comunque emessa anche nella composizione (in tal caso si sostituisce '`\kill`' con '`\\`'); inoltre, è possibile modificare gli stop di tabulazione con un'altra riga contenente comandi '`\=`'. Segue la descrizione dei comandi più importanti che riguardano l'ambiente '**tabular**':

Comando	Descrizione
<code>\=</code>	fissa uno stop di tabulazione nella posizione in cui si trova;
<code>\></code>	salta allo stop di tabulazione successivo;
<code>\kill</code>	prende in considerazione i comandi che fissano gli stop di tabulazione, ma non mostra la riga corrispondente nella composizione finale;
<code>\pushtabs</code> <code>\poptabs</code>	salva e ripristina le posizioni degli stop di tabulazione, quando si vuole avere una suddivisione diversa in un ambito temporaneo della tabella.

In realtà, i comandi specifici dell'ambiente **'tabbing'** sono molti di più; in particolare, la sovrapposizione di accenti alle lettere funziona in modo diverso dal solito. Onde evitare di creare confusione per questo motivo, si preferisce mostrare un modo alternativo di usare l'ambiente **'tabbing'**, allo scopo di aggirare questo inconveniente:

```
\begin{tabbing}
testo [\= testo]... \kill
\parbox{\textwidth}{testo} [\> \parbox{\textwidth}{testo}]... \\  

[\parbox{\textwidth}{testo} [\> \parbox{\textwidth}{testo}]
... \\  

...
\end{tabbing}
```

In pratica, si inserisce il testo di ogni cella dentro l'ambiente definito dal comando **'\parbox'**, che ripristina le condizioni di funzionamento normale.

Per definire in modo preciso gli stop di tabulazione, ci si può avvalere del comando **'\hspace'**, il cui scopo è quello di produrre uno spazio orizzontale ben definito:

```

\begin{tabbing}
\hspace*{largh_col} [\>= \hspace*{largh_col}] ... \kill
\parbox{\textwidth}{testo} ←
↔ [\> \parbox{\textwidth}{testo}] ... \\\
[\parbox{\textwidth}{testo} ←
↔ [\> \parbox{\textwidth}{testo}] ... \\\]
...
\end{tabbing}

```

L'asterisco che appare nel comando '`\hspace`' è un'opzione necessaria a evitare che lo spazio finale venga eliminato (diversamente non si otterrebbe l'effetto desiderato). Viene ripreso l'esempio iniziale, modificato con le tecniche appena descritte, aggiungendo una colonna ulteriore per dei commenti:

```

\begin{tabbing}
\hspace*{3cm} \>= \hspace*{2cm} \>= \hspace*{3cm} \>= \hspace*{4cm} \kill

\parbox{\textwidth}{nero} \> \parbox{\textwidth}{0}
 \> \parbox{\textwidth}{zero} \\\
\parbox{\textwidth}{marrone} \> \parbox{\textwidth}{1}
 \> \parbox{\textwidth}{uno} \\\
\parbox{\textwidth}{rosso} \> \parbox{\textwidth}{2}
 \> \parbox{\textwidth}{due} \\\
\parbox{\textwidth}{arancio} \> \parbox{\textwidth}{3}
 \> \parbox{\textwidth}{tre} \\\
\parbox{\textwidth}{giallo} \> \parbox{\textwidth}{4}
 \> \parbox{\textwidth}{quattro} \\\
\parbox{\textwidth}{verde} \> \parbox{\textwidth}{5}
 \> \parbox{\textwidth}{cinque} \\\
\parbox{\textwidth}{blu} \> \parbox{\textwidth}{6}
 \> \parbox{\textwidth}{sei} \\\
\parbox{\textwidth}{viola} \> \parbox{\textwidth}{7}
 \> \parbox{\textwidth}{sette} \\\
\parbox{\textwidth}{grigio} \> \parbox{\textwidth}{8}
 \> \parbox{\textwidth}{otto} \\\

```

```

\parbox{\textwidth}{bianco} \> \parbox{\textwidth}{9}
  \> \parbox{\textwidth}{nove} \\\

\parbox{\textwidth}{argento} \> \parbox{\textwidth}{10 \%}
  \> \> \parbox{\textwidth}{bla bla bla bla bla bla bla bla bla bla} \\\
\parbox{\textwidth}{oro} \> \parbox{\textwidth}{5 \%} \\\
  \> \parbox{\textwidth}{2 \%} \\\
  \> \parbox{\textwidth}{1 \%} \\\

\end{tabbing}

```

Per il momento, le tecniche mostrate non risolvono ancora il problema del traboccamento del testo da una colonna a quella successiva. A questo proposito, nell'esempio si può osservare che il commento nella quarta colonna, per la quale si prevede una larghezza di 4 cm, va oltre lo spazio a disposizione.

nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

Per ovviare a questo inconveniente, occorre controllare meglio il comando '**\parbox**', specificando esattamente la larghezza, nello stesso modo usato per la definizione degli stop di tabulazione:

```

\begin{tabbing}
\hspace*{3cm} \= \hspace*{2cm} \= \hspace*{3cm}
  \= \hspace*{4cm} \kill

\parbox{3cm}{nero} \> \parbox{2cm}{0}

```

```

    \> \parbox{3cm}{zero} \\
\parbox{3cm}{marrone} \> \parbox{2cm}{1}
    \> \parbox{3cm}{uno} \\
\parbox{3cm}{rosso} \> \parbox{2cm}{2}
    \> \parbox{3cm}{due} \\
\parbox{3cm}{arancio} \> \parbox{2cm}{3}
    \> \parbox{3cm}{tre} \\
\parbox{3cm}{giallo} \> \parbox{2cm}{4}
    \> \parbox{3cm}{quattro} \\
\parbox{3cm}{verde} \> \parbox{2cm}{5}
    \> \parbox{3cm}{cinque} \\
\parbox{3cm}{blu} \> \parbox{2cm}{6}
    \> \parbox{3cm}{sei} \\
\parbox{3cm}{viola} \> \parbox{2cm}{7}
    \> \parbox{3cm}{sette} \\
\parbox{3cm}{grigio} \> \parbox{2cm}{8}
    \> \parbox{3cm}{otto} \\
\parbox{3cm}{bianco} \> \parbox{2cm}{9}
    \> \parbox{3cm}{nove} \\

\parbox{3cm}{argento} \> \parbox{2cm}{10 \%}
    \> \> \parbox{4cm}{bla bla bla bla bla bla bla
                    bla bla bla bla} \\
\parbox{3cm}{oro} \> \parbox{2cm}{5 \%} \\
                    \> \parbox{2cm}{2 \%} \\
                    \> \parbox{2cm}{1 \%} \\
\end{tabbing}

```

In questo modo, finalmente, il testo che eccede lo spazio a disposizione viene suddiviso automaticamente in più righe; inoltre, è possibile controllare la posizione verticale del testo nella cella, attraverso le opzioni di ‘**\parbox**’.

nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla
oro	5 %		bla bla bla bla
	2 %		
	1 %		

L'ultimo problema che vale la pena di prendere in considerazione in questo tipo di ambiente, è la definizione di tabulazioni relative all'ampiezza del testo. Ciò si ottiene antepoendo un numero davanti al comando che restituisce già questo valore. Per esempio, `'0.5\textwidth'` rappresenta esattamente la metà dell'ampiezza orizzontale a disposizione. Viene riproposto l'esempio consueto con l'indicazione di ampiezze relative:

```
\begin{tabbing}
\hspace*{0.2\textwidth} \= \hspace*{0.1\textwidth}
\= \hspace*{0.2\textwidth}
\= \hspace*{0.8\textwidth} \kill

\parbox{0.2\textwidth}{nero} \> \parbox{0.1\textwidth}{0}
\> \parbox{0.2\textwidth}{zero} \\\
\parbox{0.2\textwidth}{marrone} \> \parbox{0.1\textwidth}{1}
\> \parbox{0.2\textwidth}{uno} \\\
\parbox{0.2\textwidth}{rosso} \> \parbox{0.1\textwidth}{2}
\> \parbox{0.2\textwidth}{due} \\\
\parbox{0.2\textwidth}{arancio} \> \parbox{0.1\textwidth}{3}
```

```
\> \parbox{0.2\textwidth}{tre} \\
\parbox{0.2\textwidth}{giallo} \> \parbox{0.1\textwidth}{4}
\> \parbox{0.2\textwidth}{quattro} \\
\parbox{0.2\textwidth}{verde} \> \parbox{0.1\textwidth}{5}
\> \parbox{0.2\textwidth}{cinque} \\
\parbox{0.2\textwidth}{blu} \> \parbox{0.1\textwidth}{6}
\> \parbox{0.2\textwidth}{sei} \\
\parbox{0.2\textwidth}{viola} \> \parbox{0.1\textwidth}{7}
\> \parbox{0.2\textwidth}{sette} \\
\parbox{0.2\textwidth}{grigio} \> \parbox{0.1\textwidth}{8}
\> \parbox{0.2\textwidth}{otto} \\
\parbox{0.2\textwidth}{bianco} \> \parbox{0.1\textwidth}{9}
\> \parbox{0.2\textwidth}{nove} \\

\parbox{0.2\textwidth}{argento}
\> \parbox{0.1\textwidth}{10 \%}
\> \> \parbox{0.3\textwidth}{bla bla bla bla bla bla bla
                          bla bla bla bla} \\

\parbox{0.2\textwidth}{oro}
\> \parbox{0.1\textwidth}{5 \%} \\
\> \parbox{0.1\textwidth}{2 \%} \\
\> \parbox{0.1\textwidth}{1 \%} \\
\end{tabbing}
```

nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

Questo tipo di tabella, ottenuta attraverso degli stop di tabulazione, è fatta per poter essere separata anche in più pagine; inoltre, non si adatta facilmente all’inserimento di bordi attorno alle celle. Tuttavia, per la gestione dei bordi si preferisce di solito l’ambiente ‘**tabular**’.

50.11.2 Tabella fatta di colonne e righe

«

Anche se in pratica si tratta sempre della stessa cosa, l’ambiente ‘**tabular**’ non fa riferimento a degli stop di tabulazione, ma a delle colonne, definite contestualmente alla dichiarazione iniziale dell’ambiente:

```
\begin{tabular}{definizione_delle_colonne}
testo & testo [& testo] ... \\
[testo & testo [& testo] ... \\
...
\end{tabular}
```


Il modello appena mostrato è semplificato rispetto alle possibilità complessive dell'ambiente `'tabular'`. Come si può intuire, il simbolo `'&'` viene usato per identificare la fine di una colonna e l'inizio di quella successiva; inoltre, come già avveniva per l'ambiente `'tabbing'`, si usa il comando `'\\'` per dichiarare la fine di una riga della tabella.

La definizione delle colonne avviene attraverso una stringa particolare che può essere anche molto articolata. Per cominciare in modo semplice, si può usare una lettera per ogni colonna, dove la lettera ha un significato specifico:

Lettera	Descrizione
l	rappresenta una colonna il cui contenuto deve essere allineato a sinistra;
r	rappresenta una colonna il cui contenuto deve essere allineato a destra;
c	rappresenta una colonna il cui contenuto deve essere allineato al centro.

Inoltre, esiste la possibilità di segnalare l'inserzione di una linea verticale alle estremità oppure tra una colonna e l'altra:

Simbolo	Descrizione
	rappresenta una separazione attraverso una linea verticale.

Per esempio, la tabella che viene dichiarata nel modo seguente, indica la presenza di tre colonne, dove la prima e la terza ha il testo allineato a sinistra, mentre quella centrale ha il testo allineato a destra; inoltre, definisce un bordo verticale sinistro, un bordo destro e un bordo tra la prima e la seconda colonna.

```
\begin{tabular}{|l|r|}
```

Viene riproposto un esempio già mostrato a proposito dell'ambiente **'tabbing'**, secondo questa definizione di colonne:

```
\begin{tabular}{|l|r|}
nero           & 0           & zero         & \\
marrone       & 1           & uno          & \\
rosso         & 2           & due          & \\
arancio       & 3           & tre          & \\
giallo        & 4           & quattro      & \\
verde         & 5           & cinque       & \\
blu           & 6           & sei          & \\
viola         & 7           & sette        & \\
grigio        & 8           & otto         & \\
bianco        & 9           & nove         & \\
argento       & 10 \%      &              & \\
oro           & 5 \%       &              & \\
              & 2 \%       &              & \\
              & 1 \%       &              & \\
\end{tabular}
```

nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre
giallo	4	quattro
verde	5	cinque
blu	6	sei
viola	7	sette
grigio	8	otto
bianco	9	nove
argento	10 %	
oro	5 %	
	2 %	
	1 %	

Come si vede, non è stata dichiarata l'ampiezza delle colonne, pertanto questa viene determinata automaticamente, prendendo in considerazione l'ampiezza della cella che contiene il testo più lungo, anche se questo può produrre una tabella più larga dello spazio a disposizione nel corpo del documento.

Sempre rimanendo a un livello realizzativo molto semplice, si possono inserire delle linee orizzontali, attraverso il comando `\hline`, come nella variante seguente, in cui si aggiunge anche un'intestazione molto semplice:

```
\begin{tabular}{|l|l|r|l|}
\hline
Colore          & Numero          & & \\
\hline
nero            & 0               & zero & \\
marrone        & 1               & uno  & \\
rosso          & 2               & due  & \\
\end{tabular}
```

```

arancio      & 3      & tre      \\
giallo       & 4      & quattro  \\
verde        & 5      & cinque   \\
blu          & 6      & sei      \\
viola        & 7      & sette    \\
grigio       & 8      & otto     \\
bianco       & 9      & nove     \\
argento      & 10 \% &          \\
oro          & 5 \% &          \\
            & 2 \% &          \\
            & 1 \% &          \\
\hline
\end{tabular}

```

Colore	Numero
nero	0 zero
marrone	1 uno
rosso	2 due
arancio	3 tre
giallo	4 quattro
verde	5 cinque
blu	6 sei
viola	7 sette
grigio	8 otto
bianco	9 nove
argento	10 %
oro	5 %
	2 %
	1 %

Per quanto visto fino a questo momento, si intende che non si possono ottenere celle in cui il testo utilizzi più di una riga. Tuttavia, come

già mostrato a proposito dell'ambiente `'tabbing'`, si può sfruttare il comando `'\parbox'`:

```
\documentclass{article}
\usepackage[italian]{babel}
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
\frenchspacing
\pagestyle{empty}
\setlength{\textwidth}{14cm}
\begin{document}

\begin{tabular}{|l|r|l|}
\hline
\parbox{0.2\textwidth}{Colore}
& \parbox{0.1\textwidth}{Numero} & & \\
\hline
\parbox{0.2\textwidth}{nero} & \parbox{0.1\textwidth}{0}
& \parbox{0.2\textwidth}{zero} & \\
\parbox{0.2\textwidth}{marrone} & \parbox{0.1\textwidth}{1}
& \parbox{0.2\textwidth}{uno} & \\
\parbox{0.2\textwidth}{rosso} & \parbox{0.1\textwidth}{2}
& \parbox{0.2\textwidth}{due} & \\
\parbox{0.2\textwidth}{arancio} & \parbox{0.1\textwidth}{3}
& \parbox{0.2\textwidth}{tre} & \\
\parbox{0.2\textwidth}{giallo} & \parbox{0.1\textwidth}{4}
& \parbox{0.2\textwidth}{quattro} & \\
\parbox{0.2\textwidth}{verde} & \parbox{0.1\textwidth}{5}
& \parbox{0.2\textwidth}{cinque} & \\
\parbox{0.2\textwidth}{blu} & \parbox{0.1\textwidth}{6}
& \parbox{0.2\textwidth}{sei} & \\
\parbox{0.2\textwidth}{viola} & \parbox{0.1\textwidth}{7}
& \parbox{0.2\textwidth}{sette} & \\
\parbox{0.2\textwidth}{grigio} & \parbox{0.1\textwidth}{8}
& \parbox{0.2\textwidth}{otto} & \\
\end{tabular}
```

```

\parbox{0.2\textwidth}{bianco} & \parbox{0.1\textwidth}{9}
& \parbox{0.2\textwidth}{nove} & \\

\parbox{0.2\textwidth}{argento}
& \parbox{0.1\textwidth}{10 \%}
& & \parbox{0.3\textwidth}{bla bla bla bla bla bla bla
& & & bla bla bla bla} \\

\parbox{0.2\textwidth}{oro}
& \parbox{0.1\textwidth}{5 \%}
& & \\
& & & \parbox{0.1\textwidth}{2 \%}
& & \\
& & & \parbox{0.1\textwidth}{1 \%}
& & \\

\hline
\end{tabular}

\end{document}

```

Colore	Numero		
nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

Per ottenere un risultato molto simile a questo, si potrebbe usare una dichiarazione particolare nella definizione delle colonne:

```
\begin{tabular}{|p{0.2\textwidth}|←
→p{0.1\textwidth}p{0.2\textwidth}|p{0.3\textwidth}|}
```

In tal modo si semplifica l'indicazione dei dati nelle celle:

```
\begin{tabular}{|p{0.2\textwidth}|p{0.1\textwidth}
p{0.2\textwidth}|p{0.3\textwidth}|}
\hline
Colore & Numero & & & \\
\hline
nero & 0 & zero & & \\
marrone & 1 & uno & & \\
rosso & 2 & due & & \\
arancio & 3 & tre & & \\
giallo & 4 & quattro & & \\
verde & 5 & cinque & & \\
blu & 6 & sei & & \\
\hline
```

```

viola      & 7      & sette      &          \\
grigio    & 8      & otto      &          \\
bianco    & 9      & nove      &          \\
argento   & 10 \% &          & bla bla bla bla
          &          &          & bla bla bla bla
          &          &          & bla bla bla \\
oro       & 5 \% &          &          \\
          & 2 \% &          &          \\
          & 1 \% &          &          \\
\hline
\end{tabular}

```

Colore	Numero		
nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

Come si può intuire, quella ‘**p{ }**’ è un modo per dire che il contenuto delle celle deve essere racchiuso all’interno del comando ‘**\parbox**’ e il funzionamento è molto simile.

Oltre alla possibilità di usare il comando ‘**\hline**’ per ottenere una

linea orizzontale che divida le righe della tabella, si può ottenere una linea che intervenga solo nell'ambito di alcune colonne, attraverso il comando '`\cline`'. Si osservi l'esempio seguente che è solo una piccola variante dell'ultimo già presentato:

```

\begin{tabular}{|p{0.2\textwidth}|p{0.1\textwidth}
p{0.2\textwidth}|p{0.3\textwidth}||}
\hline
Colore          & Numero &          &          & \\\
\hline
nero            & 0      & zero    &          & \\\
marrone        & 1      & uno     &          & \\\
...
\cline{1-3}
argento        & 10 \%  &          &          & bla bla bla
                                   bla bla bla
                                   bla bla bla
                                   bla bla \\\
oro            & 5 \%   &          &          & \\\
\cline{2-4}
               & 2 \%   &          &          & \\\
               & 1 \%   &          &          & \\\
\hline
\end{tabular}

```

In pratica, '`\cline{1-3}`' genera una linea orizzontale che parte dall'inizio della prima colonna, fino alla fine della terza colonna; nello stesso modo, '`\cline{2-4}`' genera una linea dalla seconda colonna fino alla quarta inclusa:⁴

Colore	Numero		
nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

50.11.3 Tabella a cavallo di più pagine

«

L'ambiente **'tabular'** genera un blocco che non può essere diviso tra le pagine; per ottenere una tabella che può continuare nelle pagine successive, si può usare l'ambiente **'longtable'**, che però richiede un pacchetto esterno con lo stesso nome. Il funzionamento generale dell'ambiente **'longtable'** è molto simile a quello di **'tabular'**, con l'aggiunta di qualche comando che consente di individuare un'intestazione e un piede da ripetere quando la tabella si interrompe e riprende nella pagina successiva:

Comando	Descrizione
<code>\usepackage{longtable}</code>	dichiara l'utilizzo del pacchetto necessario alla gestione delle tabelle «lunghe»;

Comando	Descrizione
<code>\begin{longtable}{<i>descrizione_colonne</i>}</code> ... <code>\end{longtable}</code>	dichiara l'ambiente di una tabella lunga;
<code>\endhead</code>	dichiara la fine delle righe da usare come intestazione;
<code>\endfoot</code>	dichiara la fine delle righe da usare alla base della tabella.

In generale, una volta dichiarato l'ambiente, si inseriscono le righe da usare come intestazione e subito dopo quelle da usare alla base della tabella; infine si inseriscono le righe del corpo della tabella.

Per il funzionamento dell'ambiente '**longtable**' non è indispensabile individuare delle righe di intestazione e di conclusione della tabella; tuttavia, per ottenere un risultato gradevole al cambio pagina, conviene sfruttare questa caratteristica.

L'esempio seguente è solo una piccola variante di un altro già apparso a proposito dell'ambiente '**tabular**'. Si osservi in particolare l'uso dei comandi '**\endhead**' e '**\endfoot**':

```
...
\usepackage{longtable}
...
\begin{document}
...
\begin{longtable}{|p{0.2\textwidth}|p{0.1\textwidth}
p{0.2\textwidth}|p{0.3\textwidth}||}

```

```

\hline
Colore          & Numero &          &          \\
\hline
\endhead
\hline
\endfoot
nero            & 0      & zero  &          \\
marrone        & 1      & uno   &          \\
...
                & 1  \%  &       &          \\
\end{longtable}

```

Come si vede nell'esempio, ciò che viene messo alla base della tabella è semplicemente un comando '**\hline**', per ottenere una linea orizzontale quando la tabella si interrompe; così facendo, alla fine della tabella non viene aggiunto un comando '**\hline**', perché altrimenti i due si sommerebbero:

Colore	Numero		
nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

Le tabelle ottenute con l'ambiente `'longtable'` hanno degli inconvenienti. Quello più appariscente riguarda l'incapacità di tenere in considerazione le note a piè di pagina, le quali vengono tranquillamente sovrascritte da una tabella che scorre anche sulla pagina successiva.

50.12 LaTeX: figure

Esistono diversi modi di realizzare dei disegni o di importare delle immagini esterne con LaTeX. Probabilmente, la tecnica più comune è quella di incorporare un'immagine o un disegno realizzato con strumenti interattivi esterni, ma è comunque importante conoscere più possibilità a disposizione, anche se non nel dettaglio.

50.12.1 Disegnare con LaTeX

«

LaTeX offre un ambiente particolare, ‘**picture**’, nel quale è possibile realizzare dei disegni utilizzando eventualmente dei comandi specifici. In generale, si tratta soprattutto di grafici, che contengono anche del testo collocato in modo preciso. L’esempio seguente serve per dare un punto di riferimento visivo alla descrizione che viene fatta successivamente a proposito di questo ambiente:

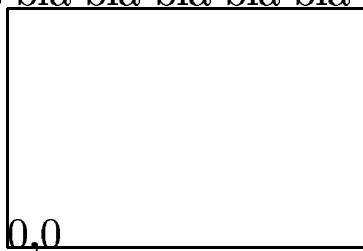
```
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla
```

```
\setlength{\unitlength}{1cm}
\begin{picture}(3,2)(0,0)
\put(0,0){0,0}
\put(0,0){\line(1,0){3}}
\put(0,2){\line(1,0){3}}
\put(0,0){\line(0,1){2}}
\put(3,0){\line(0,1){2}}
\put(3,2){3,2}
\end{picture}
```

```
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla
```

Ecco il risultato della composizione. Il testo prima e dopo la figura, serve a comprendere in che modo LaTeX la allinea e quanto viene spaziata verticalmente:

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 bla bla bla bla bla bla bla bla



bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 bla bla bla bla bla bla

L'ambiente **'picture'** viene dichiarato in modo insolito rispetto agli altri comandi di LaTeX:

```
\setlength{\unitlength}{dimensione}

...

\begin{picture} (ampiezza, altezza) (coordinata_iniziale_x, ↵
↵
                 coordinata_iniziale_y)
    \put (coordinata_x, coordinata_y) {oggetto}
    ...
    ...
\end{picture}
```

In pratica, da qualche parte del documento, prima dell'uso dell'ambiente **'picture'**, occorre assegnare un valore (non elastico) alla lunghezza **'\unitlength'**; questa lunghezza diventa poi l'unità di misura delle lunghezze e delle coordinate usate nell'ambito dell'ambiente **'picture'**. Naturalmente, il valore di **'\unitlength'** può essere cambiato successivamente, prima di altre figure.

Come si può vedere, l'ambiente `'picture'` utilizza degli argomenti tra parentesi tonde, che non si trovano altrimenti in LaTeX, ma la cosa più interessante è che la seconda parentesi tonda è facoltativa.⁵

La prima coppia di numeri tra parentesi tonde serve a delimitare un rettangolo ideale, all'interno del quale si intende collocare la figura. Nel caso dell'esempio, dal momento che `'\unitlength'` è di 1 cm, si tratta di un rettangolo largo 3 cm e alto 2 cm. La seconda coppia di numeri, opzionale, specifica le coordinate dell'angolo inferiore sinistro. Di solito si assegna a queste coordinate il valore zero, come si vede nell'esempio.

Gli «oggetti» grafici o testuali che vengono piazzati nella figura possono anche essere collocati al di fuori del rettangolo che la dovrebbe contenere, ma ovviamente, in tal modo viene meno il senso della definizione all'inizio dell'ambiente `'picture'`.

Tutto ciò che appare all'interno dell'ambiente `'picture'` viene inserito tramite il comando `'\put'`, dove, sempre tra parentesi quadre, si individuano le coordinate iniziali di un oggetto da collocare. Queste coordinate iniziali si intendono sempre relative alle coordinate iniziali del rettangolo in cui si colloca la figura.

L'esempio introduttivo, mostra un riquadro, disegnato sul bordo esterno del rettangolo che rappresenta l'area della figura. Inoltre, vengono collocate due etichette testuali, a partire dall'angolo inferiore sinistro di questo rettangolo e a partire dall'angolo superiore destro, per consentire di comprendere in che modo viene collocato il testo. In effetti, si può vedere che quanto scritto a partire dall'ango-

lo superiore destro si trova in pratica al di fuori del rettangolo della figura.

All'interno del controllo di '`\put`' possono essere usati diversi comandi specifici, con i quali si realizzano facilmente linee, riquadri, ovali e cerchi. Viene mostrato un elenco parziale di questi.

Comando	Descrizione
<code>\circle*{<i>diametro</i>}</code>	Disegna un cerchio con centro nelle coordinate iniziali, del diametro indicato, espresso in quantità di ' <code>\unitlength</code> '. Se si usa l'asterisco, si ottiene un cerchio pieno (nero).
<code>\dashbox{<i>lunghezza_del_tratteggio</i>} ←</code> <code>↪ (<i>ampiezza</i> , <i>altezza</i>) ←</code> <code>↪ [t l b r tl bl br tr] {<i>testo</i>}</code>	Disegna un rettangolo tratteggiato, con le coordinate iniziali corrispondenti all'angolo inferiore sinistro. L'ampiezza e l'altezza sono numeri che esprimono quantità di ' <code>\unitlength</code> '. All'interno del rettangolo si può collocare del testo, contenuto nell'ultimo argomento, che di solito appare al centro, salvo il suo riposizionamento con l'uso di lettere particolari nell'argomento opzionale tra parentesi quadre.

Comando	Descrizione
<pre>\makebox (<i>ampiezza</i> , <i>altezza</i>) ↵ ↵ [t l b r tl bl br tr] {<i>testo</i>}</pre>	<p>Delimita un rettangolo, senza bordo, con le coordinate iniziali corrispondenti all'angolo inferiore sinistro. L'ampiezza e l'altezza sono numeri che esprimono quantità di '<code>\unitlength</code>'. All'interno del rettangolo si può collocare del testo, contenuto nell'ultimo argomento, che di solito appare al centro, salvo il suo riposizionamento con l'uso di lettere particolari nell'argomento opzionale tra parentesi quadre.</p>
<pre>\frame {<i>oggetto</i>}</pre>	<p>Disegna una cornice attorno all'oggetto, senza aggiungere altro spazio.</p>

Comando	Descrizione
<pre data-bbox="108 451 799 564">\line(<i>scostamento_x</i>, <i>scostamento_y</i>) ↔ ↔{<i>lunghezza</i>}</pre>	<p data-bbox="970 159 1489 833">Disegna una linea a partire dalle coordinate iniziali, passante idealmente nelle coordinate descritte tra parentesi tonde, come valore che va a sommarsi alle coordinate iniziali. Questo punto passante può essere indicato solo con valori interi, da zero a sei, e serve in pratica a descrivere l'inclinazione della linea.</p>
<pre data-bbox="108 901 847 1013">\vector(<i>scostamento_x</i>, <i>scostamento_y</i>) ↔ ↔{<i>lunghezza</i>}</pre>	<p data-bbox="970 842 1489 1048">Disegna una linea, come il comando '<code>\line</code>', aggiungendo una freccia all'estremità finale.</p>
<pre data-bbox="108 1187 655 1310">\oval(<i>ampiezza</i>, <i>altezza</i>) ↔ ↔[t l b r tl bl br tr]</pre>	<p data-bbox="970 1056 1489 1432">Disegna un ovale, oppure, se molto grande, un rettangolo smussato. Le lettere tra parentesi quadre, opzionali, consentono di mostrare solo una porzione dell'ovale.</p>

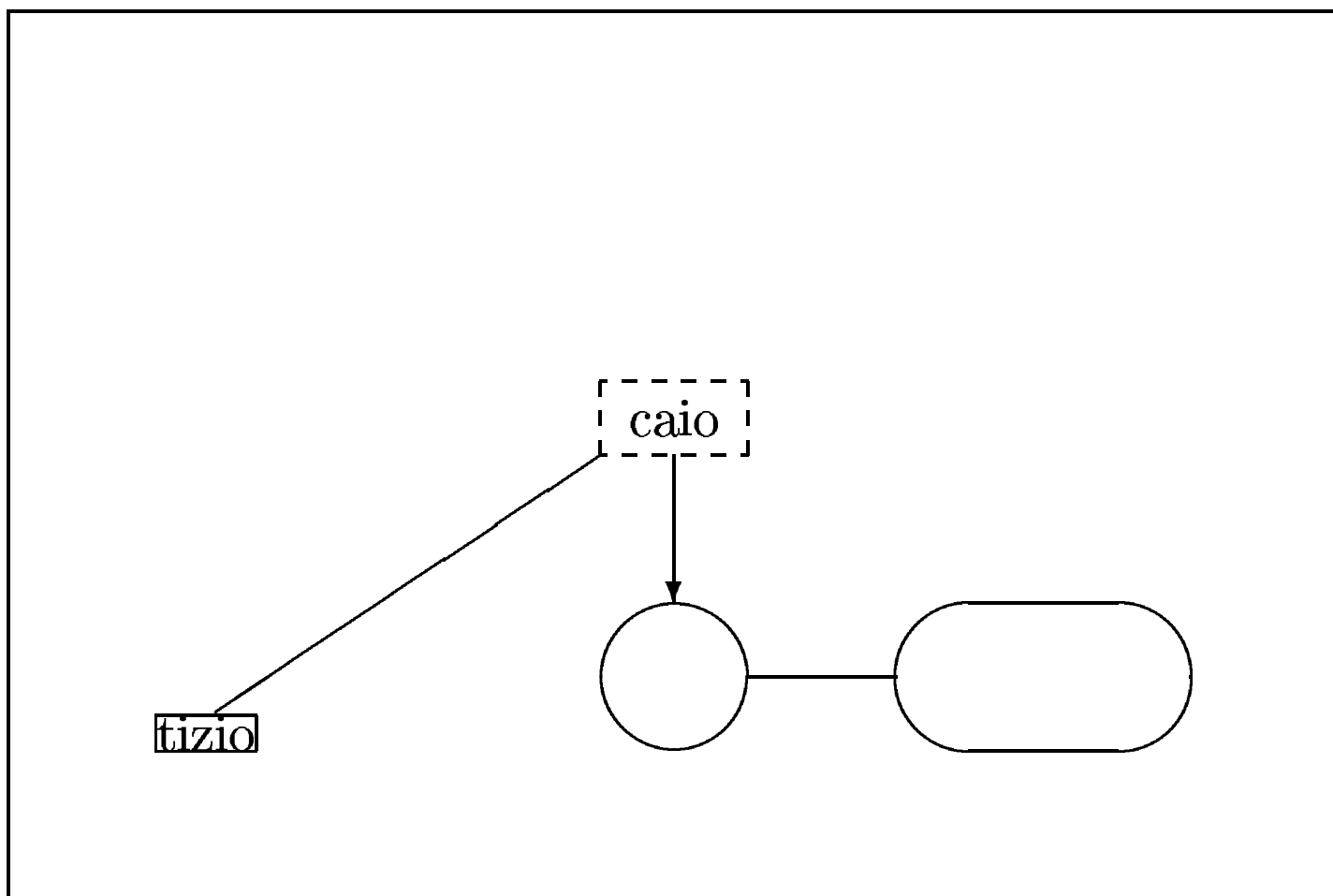
Comando	Descrizione
<code>\linethickness{<i>lunghezza</i>}</code>	<p>Consente di fissare lo spessore delle linee diritte generate da alcuni comandi. È importante osservare che la lunghezza della linea deve essere un valore positivo, espresso in modo indipendente da ‘<code>\unitlength</code>’.</p>

Viene mostrato un esempio complessivo dei comandi descritti (escluso ‘`\linethickness`’):

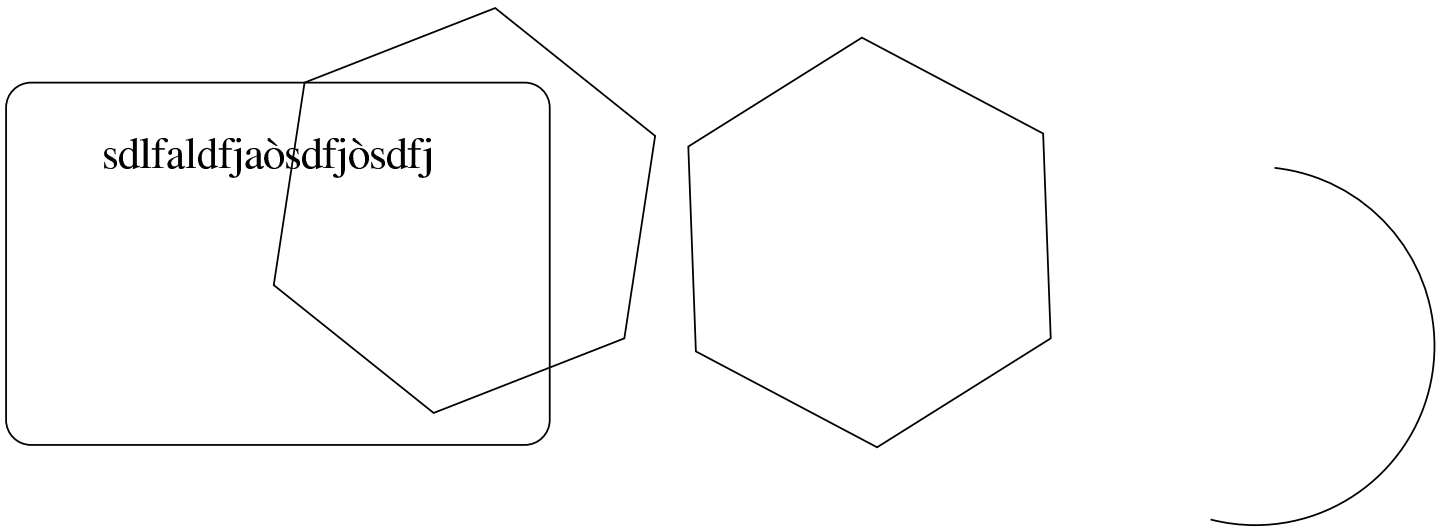
```

\setlength{\unitlength}{1cm}
\begin{picture}(9,6)(0,0)
\put(0,0){\frame{\makebox(9,6){}}}
\put(1,1){\frame{tizio}}
\put(4,3){\dashbox{0.1}(1,0.5){caio}}
\put(4,3){\line(-3,-2){2.6}}
\put(4.5,3){\vector(0,-1){1}}
\put(4.5,1.5){\circle{1}}
\put(7,1.5){\oval(2,1)}
\put(5,1.5){\line(1,0){1}}
\end{picture}

```



Come si intende, la realizzazione di disegni attraverso l'ambiente `'picture'`, può risultare impegnativo, quando il disegno diventa complesso. Tuttavia esiste la possibilità di usare XFig (sezione [29.6.4](#)), che è in grado di esportare i propri disegni in formato LaTeX, con le limitazioni grafiche necessarie. A titolo di esempio, si suppone di avere realizzato il file FIG `'disegno.fig'`, corrispondente a ciò che si può vedere sotto:



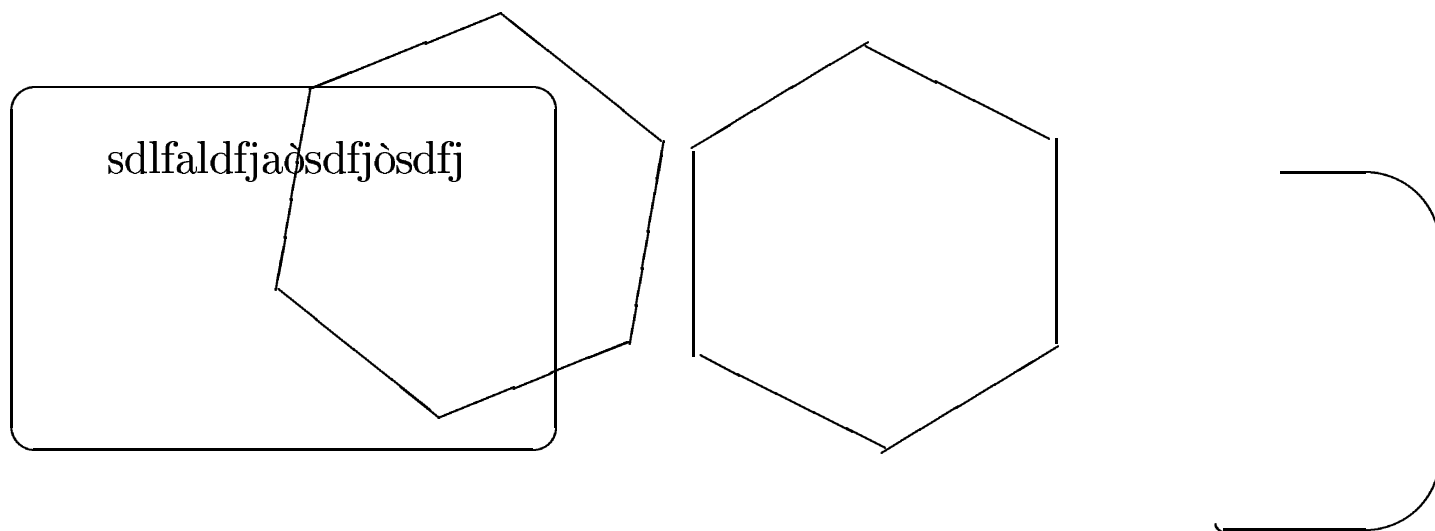
Per trasformare questo file in codice adatto a un sorgente LaTeX si può usare **‘fig2dev’** nel modo seguente, allo scopo di generare il file **‘disegno.tex’**:

```
$ fig2dev -L latex disegno.fig disegno.tex [Invio]
```

Il file che si ottiene è piuttosto complesso e probabilmente qualche comando non viene individuato correttamente da LaTeX. Per ovviare all’inconveniente, dovrebbe essere sufficiente eliminare i comandi che creano dei problemi. In generale dovrebbe trattarsi del comando **‘\color’**:

```
$ cat disegno.tex ↵
↵| sed "s/[\\]color\\\[rgb\\\[{}[0-9,]*[{}]//" ↵
↵> disegno_2.tex [Invio]
```

Ecco cosa si ottiene alla fine, con la composizione LaTeX:



50.12.2 Importazione di file EPS esterni

Un modo più semplice di inserire delle immagini nel testo realizzato con LaTeX è quello di incorporare delle immagini esterne attraverso il pacchetto ‘**epsfig**’, il quale si incorpora con il comando seguente, inserito preferibilmente nel preambolo:

```
\usepackage{epsfig}
```

Successivamente è possibile usare il comando ‘**\epsfig**’ per fare riferimento a un file EPS (*Encapsulated PostScript*) da inserire nella composizione finale. Semplificando si può indicare come nel modello sintattico seguente:

```
\epsfig{file=file_eps, height=altezza, width=larghezza, ↵  
↵angle=rotazione}
```

Il file rappresenta il percorso assoluto o relativo di un file EPS (oppure PostScript), ma senza estensione: il pacchetto ‘**epsfig**’ si attende di trovarlo con l’estensione ‘.ps’. La larghezza e l’altezza vengono espresse nell’unità di misura desiderata; infine, l’angolo di rotazione

permette di girare l'immagine, ma di solito si lascia il valore zero.

Non tutte le informazioni sono necessarie; per esempio, se si omettono le informazioni su altezza e larghezza, si fa riferimento alle dimensioni originali; nello stesso modo, se si mette solo la larghezza, si ottiene un'altezza proporzionale. In pratica, si individuano delle opzioni separate da virgole che possono essere omesse.

L'uso del pacchetto '**epsfig**' è compatibile con pdfLaTeX, il quale invece incorpora immagini in formato PNG. Questi file vengono cercati nello stesso percorso, ma con estensione '`.png`'.

In alternativa esiste anche un altro pacchetto adatto all'inserimento di immagini EPS. Si tratta di '**graphicx**' che si incorpora con il comando seguente inserito nel preambolo del sorgente LaTeX:

```
\usepackage{graphicx}
```

Successivamente si può usare il comando '**\includegraphics**':

```
\includegraphics↔  
↔[width=larghezza, height=altezza, scale=scala, angle=rotazione]↔  
↔{file_eps}
```

A differenza di quanto visto con '**\epsfig**', c'è anche la possibilità di definire una scala, attraverso un valore, dove l'unità rappresenta un rapporto pari al 100 %.

Con ‘`\includegraphics`’ esiste anche una differenza più importante rispetto a ‘`\epsfig`’: è necessario indicare anche l’estensione del file, la quale non è più determinata automaticamente. In questo modo, se si intende usare pdfLaTeX, è necessario fare riferimento a formati grafici differenti, cambiando di conseguenza l’estensione.

50.12.3 Estrazione di file EPS dal file PostScript finale

All’interno di un file PostScript, generato attraverso LaTeX e Dvips, le immagini inserite incorporando file EPS, come descritto nella sezione precedente, sono delimitate da due direttive molto precise: <<

```
%%BeginDocument: nome_file
codice_eps_della_figura
%%EndDocument
```

Sapendo questo, è possibile recuperare le immagini da un file PostScript di questo tipo attraverso un procedimento manuale, o automatico. Quello che segue è un piccolo programma Perl che legge dallo standard input il file PostScript e crea tutti i file che trova dichiarati come appena descritto (una copia di questo file dovrebbe essere disponibile anche qui: [allegati/ps-latex-picture-save.pl](#)):

```
#!/usr/bin/perl
#
my ($line)      = "";
my ($picture)  = "";
my ($save)     = 0;
#
while ($line = <STDIN>)
```

```
{
  if ($line =~ m/^\%\%BeginDocument: (\S+)\$/ )
  {
    $save = 1;
    $picture = $1;
    open (PICTURE, "> $picture");
  }
  elsif ($line =~ m/^\%\%EndDocument$/ )
  {
    close (PICTURE);
    $picture = "";
    $save = 0;
  }
  else
  {
    if ($save)
    {
      print PICTURE ("$line");
    }
  }
}
```

Supponendo che questo programma sia stato salvato nel file ‘ps-latex-picture-save’ e che sia stato reso eseguibile, si potrebbe usare come nell’esempio seguente:

```
$ ps-latex-picture-save < mio_file.ps [Invio]
```

Si osservi che questo procedimento può funzionare solo se il file PostScript di partenza è stato ottenuto direttamente da Dvips, tenendo conto che non ci devono essere state delle rielaborazioni successive, altrimenti si perdono i riferimenti descritti.

50.13 Ambienti matematici

Come avviene per TeX, anche LaTeX distingue una modalità di composizione normale, da una modalità matematica. A differenza di TeX, esistono degli ambienti speciali per la dichiarazione della modalità matematica, ma è ancora ammissibile l'uso del simbolo '\$' e di '\$\$' per individuare l'inizio e la fine degli ambienti matematici.

Molto di quanto già descritto a proposito di TeX (sezione [49.16](#)), vale ovviamente anche per la composizione matematica di LaTeX. Pertanto, alcuni dettagli non vengono descritti nuovamente.

Si tenga presente che la gestione degli ambienti matematici potrebbe essere ancora più raffinata con l'utilizzo del pacchetto '`amsmath`' (`\usepackage{amsmath}`).

50.13.1 Definizione della modalità matematica

La modalità matematica si può definire con LaTeX in diversi modi. Anche con LaTeX si distingue se la composizione matematica vada inserita nell'ambito del testo normale, oppure in un blocco indipendente. L'elenco seguente riepiloga le istruzioni necessarie a delimitare gli ambienti matematici:

Comando	Descrizione
$\begin{math} \textit{testo_matematico} \end{math}$ $\left(\textit{testo_matematico} \right)$ $\$ \textit{testo_matematico} \$$	<p>compone il testo matematico nel blocco già esistente;</p>

Comando	Descrizione
<pre>\begin{displaymath} testo_matematico \end{displaymath} \[testo_matematico\] \$\$testo_matematico\$\$</pre>	<p>compone il testo matematico in un blocco separato;</p>
<pre>\begin{equation} testo_matematico \end{equation}</pre>	<p>compone il testo matematico in un blocco separato, annotando sul bordo il numero dell'equazione.</p>

Come si vede dall'elenco, vengono offerte delle alternative per ottenere lo stesso tipo di ambiente. Vengono mostrati due esempi molto semplici, già apparsi nel capitolo riferito a TeX, che riguardano l'inserimento di una formula nel testo normale e la sua separazione in un blocco a sé stante:

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime come

```
\begin{math}{\sum {R \over T} \cdot I} \over {\sum I}
```

```
\end{math}.
```

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime come $\frac{\sum \frac{R}{T} \cdot I}{\sum I}$.

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime secondo la formula seguente:

```
\begin{displaymath}{\sum {R \over T} \cdot I}\over{\sum I}\end{displaymath}
```

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime secondo la formula seguente:

$$\frac{\sum \frac{R}{T} \cdot I}{\sum I}$$

A parte le situazioni normali, merita attenzione l'ambiente **'equation'**, con il quale si può inserire un'equazione in un blocco separato, al quale si può fare riferimento tramite un'etichetta. Si osservi l'esempio:

L'equazione `\ref{ritardo-medio}` mostra in che modo si calcola il ritardo medio, dove R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna e I è il valore della merce.

```
\begin{equation}\label{ritardo-medio}
{\sum {R \over T} \cdot I}\over{\sum I}
\end{equation}
```

L'equazione 1 mostra in che modo si calcola il ritardo medio, dove R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna e I è il valore della merce.

$$\frac{\sum \frac{R}{T} \cdot I}{\sum I} \tag{1}$$

50.13.2 Spazi orizzontali e punteggiatura

«

LaTeX si comporta nello stesso modo di TeX a proposito della spaziatura orizzontale negli ambienti matematici. In pratica, gli spazi orizzontali vengono ignorati, a meno che si usino comandi appositi per inserirli:

Sequenza	Significato
<code>\qqquad</code>	Due quadratoni.
<code>\quad</code>	Un quadratone.
<code>\langle SP \rangle</code>	Uno spazio «normale».
<code>\; </code>	Uno spazio spesso (5/8 di quadratone).
<code>\> </code>	Uno spazio medio (2/9 di quadratone).
<code>\. </code>	Uno spazio molto sottile (1/6 di quadratone).
<code>\! </code>	Uno spazio negativo (−1/6 di quadratone).

Per inserire del testo descrittivo all'interno di un ambiente matematico, lo si può delimitare in una scatola orizzontale, con il comando '`\makebox`', o più semplicemente con '`\mbox`':

```
$prima durante e dopo$\par
$prima \mbox{ durante } e dopo$\par
```

Come si può intendere, si vuole isolare e spaziare la parola «durante» rispetto al resto, mostrandola con caratteri normali.

prima durante edopo
prima durante edopo

Anche la spaziatura attorno ai simboli di punteggiatura funziona esattamente come con TeX puro e semplice. L'esempio seguente riassume i casi principali:

```
$123456.7890$\par
$123456,7890$\par
$123456{,}7890$\par
$123456;7890$\par
$123456:7890$\par
$123456:=7890$\par
$123456\colon 7890$\par
```

```
123456.7890
123456,7890
123456,7890
123456;7890
123456 : 7890
123456 := 7890
123456:7890
```

50.13.3 Caratteri e simboli

LaTeX è più ricco di simboli rispetto a TeX, se si utilizza il pacchetto **'latexsym'**:

```
\usepackage{latexsym}
```

Le tabelle che seguono riepilogano i simboli a disposizione, inclusi quelli provenienti dal pacchetto **'latexsym'**.

Tabella 50.282. Accenti nell'ambiente matematico.

Simbolo	Codice	Annotazioni
\acute{o}	<code>\acute{o}</code>	Accento acuto matematico.
\grave{o}	<code>\grave{o}</code>	Accento grave matematico.
\hat{o}	<code>\hat{o}</code>	Accento circonflesso matematico.
\ddot{o}	<code>\ddot{o}</code>	
\tilde{o}	<code>\tilde{o}</code>	
\bar{o}	<code>\bar{o}</code>	
\dot{o}	<code>\dot{o}</code>	
\breve{o}	<code>\breve{o}</code>	
\check{o}	<code>\check{o}</code>	
\vec{o}	<code>\vec{o}</code>	
\imath	<code>\imath</code>	Da usare per aggiungere un accento.
\jmath	<code>\jmath</code>	Da usare per aggiungere un accento.
\widehat{abc}	<code>\widehat{abc}</code>	
\widetilde{abc}	<code>\widetilde{abc}</code>	

Tabella 50.283. Lettere greche.

Simbolo	Codice	Simbolo	Codice	Simbolo	Codice	Nome
α	<code>\alpha</code>			A	<code>\textrm{A}</code>	Alfa
β	<code>\beta</code>			B	<code>\textrm{B}</code>	Beta
γ	<code>\gamma</code>			Γ	<code>\Gamma</code>	Gamma
δ	<code>\delta</code>			Δ	<code>\Delta</code>	Delta
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>	E	<code>\textrm{E}</code>	Epsilon
ζ	<code>\zeta</code>			Z	<code>\textrm{Z}</code>	Zeta
η	<code>\eta</code>			H	<code>\textrm{H}</code>	Eta
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>	Θ	<code>\Theta</code>	Theta
ι	<code>\iota</code>			I	<code>\textrm{I}</code>	Iota
κ	<code>\kappa</code>			K	<code>\textrm{K}</code>	Kappa
λ	<code>\lambda</code>			Λ	<code>\Lambda</code>	Lambda
μ	<code>\mu</code>			M	<code>\textrm{M}</code>	Mu
ν	<code>\nu</code>			N	<code>\textrm{N}</code>	Nu
ξ	<code>\xi</code>			Ξ	<code>\Xi</code>	Xi
\omicron	<code>\text{\texttt{O}}</code>	\omicron	<code>\text{\texttt{O}}</code>	O	<code>\text{\texttt{O}}</code>	Omicron
π	<code>\pi</code>	ϖ	<code>\varpi</code>	Π	<code>\Pi</code>	Pi
ρ	<code>\rho</code>	ϱ	<code>\varrho</code>	P	<code>\textrm{P}</code>	Rho
σ	<code>\sigma</code>	ς	<code>\varsigma</code>	Σ	<code>\Sigma</code>	Sigma
τ	<code>\tau</code>			T	<code>\textrm{T}</code>	Tau
υ	<code>\upsilon</code>			Υ	<code>\Upsilon</code>	Upsilon
ϕ	<code>\phi</code>	φ	<code>\varphi</code>	Φ	<code>\Phi</code>	Phi
χ	<code>\chi</code>			X	<code>\textrm{X}</code>	Chi
ψ	<code>\psi</code>			Ψ	<code>\Psi</code>	Psi
ω	<code>\omega</code>			Ω	<code>\Omega</code>	Omega

Tabella 50.284. Simboli matematici comuni.

Simbolo	Codice	Annotazioni
\backslash	<code>\backslash\$</code>	Barra obliqua inversa.
\aleph	<code>\aleph\$</code>	
\hbar	<code>\hbar\$</code>	
ℓ	<code>\ell\$</code>	
\wp	<code>\wp\$</code>	
\Re	<code>\Re\$</code>	
\Im	<code>\Im\$</code>	
∂	<code>\partial\$</code>	Derivata parziale.
∞	<code>\infty\$</code>	Infinito.
$'$	<code>\prime\$</code>	Primo.
$'$	<code>'\$</code>	
\emptyset	<code>\emptyset\$</code>	
∇	<code>\nabla\$</code>	
\top	<code>\top\$</code>	
\perp	<code>\bot\$</code>	
$ $	<code> \$</code>	
$ $	<code>\vert\$</code>	Equivalente a <code> \$</code> .
$ $	<code>\ </code>	
$ $	<code>\vert\$</code>	Equivalente a <code>\ </code> .
\sphericalangle	<code>\angle\$</code>	
\triangle	<code>\triangle\$</code>	
\forall	<code>\forall\$</code>	
\exists	<code>\exists\$</code>	
\neg	<code>\neg\$</code>	Equivalente a <code>\lnot\$</code> .
\flat	<code>\flat\$</code>	
\natural	<code>\natural\$</code>	
\sharp	<code>\sharp\$</code>	
\surd	<code>\surd\$</code>	Radice senza la linea superiore.
\clubsuit	<code>\clubsuit\$</code>	Fiori.
\diamond	<code>\diamondsuit\$</code>	Quadri.
\heartsuit	<code>\heartsuit\$</code>	Cuori.
\spadesuit	<code>\spadesuit\$</code>	Picche.
\mho	<code>\mho\$</code>	Richiede il pacchetto <code>latexsym</code> .
\square	<code>\Box\$</code>	”
\diamond	<code>\Diamond\$</code>	”

Tabella 50.285. Operatori binari comuni.

Simbolo	Codice	Annotazioni
\pm	<code>\pm</code>	Più o meno.
\mp	<code>\mp</code>	Meno o più.
\cdot	<code>\cdot</code>	
\setminus	<code>\setminus</code>	
$+$	<code>+</code>	Somma.
$-$	<code>-</code>	Sottrazione.
\times	<code>\times</code>	Moltiplicazione.
\div	<code>\div</code>	Divisione.
$*$	<code>*</code>	Moltiplicazione discrezionale.
\ast	<code>\ast</code>	Asterisco.
\star	<code>\star</code>	Stella.
\diamond	<code>\diamond</code>	
\circ	<code>\circ</code>	
\bullet	<code>\bullet</code>	
\cap	<code>\cap</code>	Intersezione.
\cup	<code>\cup</code>	Unione.
\oplus	<code>\oplus</code>	
\sqcap	<code>\sqcap</code>	
\sqcup	<code>\sqcup</code>	
\triangleleft	<code>\triangleleft</code>	
\triangleright	<code>\triangleright</code>	
\wr	<code>\wr</code>	
\bigcirc	<code>\bigcirc</code>	
\bigtriangleup	<code>\bigtriangleup</code>	
\bigtriangledown	<code>\bigtriangledown</code>	
\vee	<code>\vee</code>	
\lor	<code>\lor</code>	OR logico (uguale a <code>\vee</code>).
\wedge	<code>\wedge</code>	
\land	<code>\land</code>	AND logico (uguale a <code>\wedge</code>).
\oplus	<code>\oplus</code>	
\ominus	<code>\ominus</code>	
\otimes	<code>\otimes</code>	
\oslash	<code>\oslash</code>	
\odot	<code>\odot</code>	
\dagger	<code>\dagger</code>	
\ddagger	<code>\ddagger</code>	
\amalg	<code>\amalg</code>	
\triangleleft	<code>\lhd</code>	Richiede il pacchetto <code>latexsym</code> .
\triangleright	<code>\rhd</code>	”
\triangleleft	<code>\unlhd</code>	”
\triangleright	<code>\unrhd</code>	”

Tabella 50.286. Operatori di relazione.

Simbolo	Codice	Simbolo	Codice
=	<code>=\$</code>	≠	<code>\$\$\not =</code>
		≠	<code>\$\$\neq</code>
		≠	<code>\$\$\ne</code>
≡	<code>\$\$\equiv</code>	≠	<code>\$\$\not \equiv</code>
≈	<code>\$\$\sim</code>	≈	<code>\$\$\not \sim</code>
≈	<code>\$\$\simeq</code>	≈	<code>\$\$\not \simeq</code>
≈	<code>\$\$\approx</code>	≈	<code>\$\$\not \approx</code>
≈	<code>\$\$\cong</code>	≈	<code>\$\$\not \cong</code>
∞	<code>\$\$\asymp</code>	∞	<code>\$\$\not \asymp</code>
⊗	<code>\$\$\bowtie</code>		
⋮	<code>\$\$\doteq</code>		
∥	<code>\$\$\parallel</code>		
⊥	<code>\$\$\perp</code>	⊥	<code>\$\$\not \perp</code>
<	<code>\$\$<</code>	<	<code>\$\$\not <</code>
≤	<code>\$\$\leq</code>	≤	<code>\$\$\not \leq</code>
≤	<code>\$\$\le</code>	≤	<code>\$\$\not \le</code>
>	<code>\$\$></code>	>	<code>\$\$\not ></code>
≥	<code>\$\$\geq</code>	≥	<code>\$\$\not \geq</code>
≥	<code>\$\$\ge</code>	≥	<code>\$\$\not \ge</code>
⋈	<code>\$\$\prec</code>	⋈	<code>\$\$\not \prec</code>
⋈	<code>\$\$\preceq</code>	⋈	<code>\$\$\not \preceq</code>
⋈	<code>\$\$\succ</code>	⋈	<code>\$\$\not \succ</code>
⋈	<code>\$\$\succeq</code>	⋈	<code>\$\$\not \succeq</code>
⊂	<code>\$\$\ll</code>		
⊂	<code>\$\$\gg</code>		
⊂	<code>\$\$\subset</code>	⊂	<code>\$\$\not \subset</code>
⊂	<code>\$\$\subseteq</code>	⊂	<code>\$\$\not \subseteq</code>
⊃	<code>\$\$\supset</code>	⊃	<code>\$\$\not \supset</code>
⊃	<code>\$\$\supseteq</code>	⊃	<code>\$\$\not \supseteq</code>
⊃	<code>\$\$\sqsubseteq</code>	⊃	<code>\$\$\not \sqsubseteq</code>
⊃	<code>\$\$\sqsupseteq</code>	⊃	<code>\$\$\not \sqsupseteq</code>
⊃	<code>\$\$\mid</code>		
(<code>\$\$\smile</code>		
)	<code>\$\$\frown</code>		
∈	<code>\$\$\in</code>	∉	<code>\$\$\not \in</code>
∋	<code>\$\$\ni</code>	∋	<code>\$\$\not \ni</code>
⊆	<code>\$\$\owns</code>	⊆	<code>\$\$\not \owns</code>
⊥	<code>\$\$\vdash</code>		
⊥	<code>\$\$\dashv</code>		
⊥	<code>\$\$\models</code>		
∝	<code>\$\$\propto</code>		
⊂	<code>\$\$\sqsubset</code>	⊂	<code>\$\$\not \sqsubset</code>
⊃	<code>\$\$\sqsupset</code>	⊃	<code>\$\$\not \sqsupset</code>
⊗	<code>\$\$\Join</code>		

Tabella 50.287. Frecce.

Simbolo	Codice	Simbolo	Codice
\leftarrow	<code>\leftarrow</code>	\rightarrow	<code>\rightarrow</code>
\leftarrow	<code>\gets</code>	\rightarrow	<code>\to</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Rightarrow	<code>\Rightarrow</code>
\longleftarrow	<code>\longleftarrow</code>	\longrightarrow	<code>\longrightarrow</code>
\Leftrightarrow	<code>\Longleftarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>
\lharpoonup	<code>\lharpoonup</code>	\rharpoonup	<code>\rharpoonup</code>
\lharpoonright	<code>\lharpoonright</code>	\rharpoonright	<code>\rharpoonright</code>
\rightharpoonleft	<code>\rightharpoonleft</code>		
\iff	<code>\iff</code>		
\mapsto	<code>\mapsto</code>		
\longmapsto	<code>\longmapsto</code>		
\leftrightarrow	<code>\leftrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\longleftrightarrow	<code>\longleftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>
\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\nearrow	<code>\nearrow</code>	\nearrow	<code>\nearrow</code>
\swarrow	<code>\swarrow</code>	\searrow	<code>\searrow</code>
\overleftarrow{ab}	<code>\overleftarrow{ab}</code>	\overrightarrow{ab}	<code>\overrightarrow{ab}</code>
\vec{ab}	<code>\vec{ab}</code>		
\leadsto	<code>\leadsto</code>		

Tabella 50.288. Ellissi.

Simbolo	Codice	Annotazioni
\cdots	<code>\cdots</code>	Ellissi orizzontale al centro.
\ldots	<code>\ldots</code>	Ellissi orizzontale in basso.
\vdots	<code>\vdots</code>	Ellissi verticale.
\ddots	<code>\ddots</code>	Ellissi diagonale.

Tabella 50.289. Altri simboli matematici.

Simbolo	Codice	Annotazioni
Σ	<code>\sum</code>	Sommatoria.
Π	<code>\prod</code>	
\amalg	<code>\coprod</code>	
\int	<code>\int</code>	Integrale.
\oint	<code>\oint</code>	
\cap	<code>\bigcap</code>	Intersezione.
\cup	<code>\bigcup</code>	Unione.
\sqcup	<code>\bigsqcup</code>	
\vee	<code>\bigvee</code>	
\wedge	<code>\bigwedge</code>	
\oplus	<code>\bigoplus</code>	
\oplus	<code>\biguplus</code>	
\otimes	<code>\bigotimes</code>	
\odot	<code>\bigodot</code>	

50.13.4 Delimitatori

LaTeX dispone degli stessi delimitatori di TeX, mantenendo la disponibilità dei comandi ‘`\big...`’ e ‘`\Big...`’ per gli ingrandimenti controllati. La tabella seguente serve a richiamare l’uso di questi delimitatori, tenendo conto che l’uso dei comandi ‘`\big...`’ e ‘`\Big...`’ può essere esteso anche oltre gli esempi mostrati. «

Tabella 50.290. Delimitatori.

	Codice			Codice	
()	<code>\$(\$)\$</code>		()	<code>\$(\$)\$</code>	
()	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	()	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
()	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	()	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
{ }	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	{ }	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
{ }	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	{ }	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
{ }	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	{ }	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
< >	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	< >	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
< >	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	< >	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
< >	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	< >	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	[]	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
(x)	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	($\frac{x}{y}$)	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
/	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	\	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>		<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
↑ ↓	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	↑ ↓	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>
↕	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>	↕	<code>\$(\$)\$</code>	<code>\$(\$)\$</code>

Esiste comunque la possibilità di usare comandi ‘`\big...`’ ‘`\Big...`’, senza la lettera ‘**l**’ o ‘**r**’ che specifica trattarsi di un delimitatore sinistro o destro:

	Codice		Codice
$()$	<code>\big(\$ \$\big)\$</code>	$()$	<code>\Big(\$ \$\Big)\$</code>
$()$	<code>\bigg(\$ \$\bigg)\$</code>	$()$	<code>\Bigg(\$ \$\Bigg)\$</code>

LaTeX offre anche qualche delimitatore speciale; precisamente delle parentesi graffe orizzontali, da usare sopra o sotto qualcosa:

Comando	Descrizione
<code>\overbrace{<i>testo</i>}</code>	sovrappone una parentesi graffa sopra il testo;
<code>\underbrace{<i>testo</i>}</code>	sovrappone una parentesi graffa sopra il testo.

Sono ancora validi con LaTeX i comandi ‘`\overline`’ e ‘`\underline`’, per soprallineare e sottolineare nell’ambiente matematico.

50.13.5 Modifica delle caratteristiche del carattere

Anche con LaTeX sono disponibili i comandi che permettono di controllare la dimensione del carattere negli ambienti matematici. In ordine decrescente, si tratta di: ‘`\displaystyle`’, ‘`\textstyle`’, ‘`\scriptstyle`’ e ‘`\scriptscriptstyle`’. Viene mostrato un esempio già apparso a proposito di TeX:

```


$$x + \frac{y}{x + \frac{y}{x + \frac{y}{x + \frac{y}{z}}}}$$


$$x + \frac{y}{x + \frac{y}{x + \frac{y}{x + \frac{y}{z}}}}$$


$$x + \frac{y}{x + \frac{y}{x + \frac{y}{x + \frac{y}{z}}}}$$


```

$$x + \frac{y}{x + \frac{y}{x + \frac{y}{x + \frac{y}{x + \frac{y}{z}}}}}$$

$$x + \frac{y}{x + \frac{y}{x + \frac{y}{x + \frac{y}{z}}}}$$

A differenza di TeX, LaTeX aggiunge dei comandi cumulabili, anche per la definizione dello stile e della forma del carattere matematico. Nella sezione [50.7.3](#) è già stato trattato questo argomento, comunque viene riportato nuovamente l'elenco di questi comandi:

Comando	Descrizione
<code>\mathnormal{testo}</code>	testo matematico normale;
<code>\mathrm{testo}</code>	tondo;
<code>\mathbf{testo}</code>	neretto;
<code>\mathsf{testo}</code>	lineare;
<code>\mathtt{testo}</code>	dattilografico;
<code>\mathit{testo}</code>	corsivo;
<code>\mathcal{testo}</code>	calligrafico;
<code>\mathsc{testo}</code>	maiuscoletto;

50.13.6 Espressioni varie

LaTeX consente di usare i comandi per la realizzazione di frazioni e simili, già disponibili con TeX:

- `$$\{x+y \over z}$$`

$$\frac{x + y}{z}$$

- `$$\{x \over y+z}$$`

$$\frac{x}{y + z}$$

- `$$\{a \over x} + \{b \over x} = \{a+b \over x}$$`

$$\frac{a}{x} + \frac{b}{x} = \frac{a + b}{x}$$

- `$$\{x+y \ atop z}$$`

$$\frac{x + y}{z}$$

- `$$\{x+y \ choose z}$$`

$$\binom{x + y}{z}$$

- `$$\{x+y \ above 1mm z}$$`

$$\frac{x + y}{z}$$

Inoltre, LaTeX mette a disposizione un comando aggiuntivo per le frazioni, che forse può risultare comodo in alcune situazioni:

Comando	Descrizione
<code>\frac{<i>numeratore</i>}{<i>denominatore</i>}</code>	frazione normale.

Gli apici e i pedici funzionano con LaTeX esattamente come in TeX:

- `$$10^2$$`

$$10^2$$

- `$$10^{20}$$`

$$10^{20}$$

- `$$10^{-2}$$`

$$10^{-2}$$

- `$$x^a_b$$`

$$x_b^a$$

- `$$x^{y^z}$$`

$$x^{y^z}$$

- `$\sum_{x=0}^n x^3$`

$$\sum_{x=0}^n x^3$$

- `$$\sum_{x=0}^n x^3$$`

$$\sum_{x=0}^n x^3$$

- `$$\int_0^n f(x)$$`

$$\int_0^n f(x)$$

- `$$\lim_{n \to 0} \{1 \over n\} = \infty$$`

$$\lim_{n \rightarrow 0} \frac{1}{n} = \infty$$

- `$$\lim_{n \to 0} \{1 \over n\} = \infty$$`

$$\lim_{n \rightarrow 0} \frac{1}{n} = \infty$$

Per ottenere una radice, LaTeX utilizza il comando ‘`\sqrt`’, che però si comporta in modo leggermente diverso da TeX:

Comando	Descrizione
<code>\sqrt[n]{argomento}</code>	produce la radice <i>n</i> dell’argomento indicato.

Seguono alcuni esempi:

- `$$\sqrt{x^2+y^2}$$`

$$\sqrt{x^2 + y^2}$$

- `$$\sqrt[3]{8}$$`

$$\sqrt[3]{8}$$

- `$$\sqrt{x+y \over z}$$`

$$\sqrt{\frac{x+y}{z}}$$

- `$$x^{\sqrt{y}}$$`

$$x^{\sqrt{y}}$$

LaTeX mette a disposizione gli stessi comandi di TeX per la scrittura dei nomi delle funzioni (sezione 49.17.8). Tuttavia, se si desidera scrivere il nome di una funzione in modo differente, lo si può inserire in una scatola orizzontale:

`$$\mbox{sen}(2x)$$`

$$\text{sen}(2x)$$

Per sovrapporre qualcosa si può usare il comando ‘`\stackrel`’:

`$$\stackrel{a}{B}$$`

$$\begin{matrix} a \\ B \end{matrix}$$

50.13.7 Matrici e sistemi di equazioni

«

Con LaTeX si rappresentano le matrici in modo molto simile alle tabelle:

```

\begin{array}{definizione_delle_colonne}
testo & testo [& testo] ... \\
[testo & testo [& testo] ... \\
...
\end{array}

```

La definizione delle colonne avviene, come per l'ambiente '**tabular**', attraverso una stringa composta da tre lettere possibili:

Lettera	Significato mnemonico	Descrizione
l	<i>left</i>	rappresenta una colonna il cui contenuto deve essere allineato a sinistra;
r	<i>right</i>	rappresenta una colonna il cui contenuto deve essere allineato a destra;
c	<i>center</i>	rappresenta una colonna il cui contenuto deve essere allineato al centro.

Comunque, come nell'ambiente '**tabular**', è possibile anche inserire il simbolo '|' per ottenere una linea verticale e si può usare il comando '**\hline**' per inserire una linea orizzontale.

Segue un esempio molto semplice:

```

\left (
\begin{array}{cc|c}
a & b & c \\
d & e+f & g \\
\hline
h & i & j \\
\end{array}
\right )

```

Come si può osservare, l'ambiente '**array**' crea la tabella in cui vanno inseriti gli elementi della matrice, mentre la delimitazione esterna deve essere aggiunta espressamente:

$$\left(\begin{array}{cc|c} a & b & c \\ d & e+f & g \\ \hline h & i & j \end{array} \right)$$

Trattandosi di una matrice, l'ambiente '**array**' si presta in pratica anche per realizzare un sistema di equazioni; è sufficiente regolare correttamente l'allineamento delle colonne, come nell'esempio seguente:

```

|x| =
\bigg \{
\begin{array}{r}
x & x \geq 0 \\
-x & x \leq 0
\end{array}
\}

```

$$|x| = \begin{cases} x & x \geq 0 \\ -x & x \leq 0 \end{cases}$$

L'ambiente '**array**' può essere usato in pratica nello stesso modo di '**tabular**', consentendo così la realizzazione di schemi più complessi di quanto mostrato qui. Eventualmente si consulti la sezione [50.11.2](#).

Tuttavia, esiste un ambiente speciale per elencare delle equazioni allineate a cui poter fare riferimento; si tratta di '**eqnarray**':

```
\begin{eqnarray}
testo & & testo & & testo & \\\
testo & & testo & & testo & \\\
...
\end{eqnarray}
```

In pratica, si tratta di una tabella in cui si presuppone la presenza di tre colonne, allineate rispettivamente a sinistra, al centro e a sinistra. Sarebbe semplicissimo realizzare questo con l'ambiente '**array**', ma la differenza sta nel fatto che ogni riga viene numerata in qualità di equazione. Si osservi l'esempio, in cui si inseriscono anche delle etichette che poi consentono di farvi riferimento:

```
\begin{eqnarray}
x & = & 2y & \label{equazione_xy} & \\\
y & = & 3z & \label{equazione_yz} & \\\
\end{eqnarray}

L'equazione \ref{equazione_xy} dice che... mentre
l'equazione \ref{equazione_yz} fa riferimento a...
```

Si osservi il fatto che l'ultima riga della matrice non è terminata con il simbolo '\', perché diversamente si otterrebbe un'equazione in più, numerata, anche se invisibile:

$$x = 2y \tag{1}$$

$$y = 3z \tag{2}$$

L'equazione 1 dice che... mentre l'equazione 2 fa riferimento a...

Ancora più importante è notare che l'ambiente `'eqnarray'` viene dichiarato in un contesto normale, ma delimita un ambiente matematico particolare.

Eventualmente, per evitare di numerare una certa riga, si può usare il comando `'\nonumber'`; inoltre, se si aggiunge un asterisco al nome dell'ambiente, non viene mostrata la numerazione di alcuna equazione:

```
\begin{eqnarray*}
x & = & 2y \label{equazione_xy} \\
y & = & 3z \label{equazione_yz}
\end{eqnarray*}
```

L'equazione `\ref{equazione_xy}` dice che... mentre l'equazione `\ref{equazione_yz}` fa riferimento a...

$$x = 2y$$

$$y = 3z$$

L'equazione 1 dice che... mentre l'equazione 1 fa riferimento a...

50.13.8 Teoremi, corollari e dichiarazioni simili



LaTeX ha una gestione sofisticata delle dichiarazioni riconducibili a teoremi, corollari e simili. Si parte dalla dichiarazione di un ambiente, al quale si associa un contatore, in modo tale da non porre limitazioni al genere di dichiarazioni che si vogliono gestire.

Inizialmente si definisce il tipo di contesto a cui appartiene un certo gruppo di dichiarazioni:

```
\newtheorem{ambiente}{Titolo}[contatore_di_riferimento]
```

```
\newtheorem{ambiente}[ambiente_collegato]{Titolo}
```

La seconda delle due forme permette di definire un ambiente la cui numerazione deve seguire quella di un altro già definito. In questo senso, nel campo indicato come *ambiente_collegato*, va messo il nome di quello già esistente.

Per cominciare, si osservi l'esempio seguente, senza l'indicazione di alcun contatore di riferimento:

```
\newtheorem{teorema}{Teorema}
```

In questo modo si dichiara l'ambiente '**teorema**', con il titolo «Teorema», assieme al contatore '**teorema**' (a cui si accede facilmente con il comando '**\thetheorem**', creato implicitamente), usato in modo trasparente.

Nel dichiarare un ambiente di questo tipo, occorre evitare di scegliere nomi già usati da LaTeX.

Dopo la definizione dell'ambiente '**teorema**', si può usare in pratica questo ambiente; per esempio così:

```

\begin{teorema}
In teoria, non c'è differenza tra pratica e teoria.
\end{teorema}

\begin{teorema}
In pratica, la teoria è diversa dalla pratica.
\end{teorema}

```

Dal risultato della composizione si può osservare che il titolo, ovvero la parola «Teorema», viene messo in evidenza con un carattere nero, assieme al numero del contatore; subito dopo appare il testo del teorema in un carattere corsivo:

Teorema 1 *In teoria, non c'è differenza tra pratica e teoria.*

Teorema 2 *In pratica, la teoria è diversa dalla pratica.*

Nella definizione dell'ambiente **'teorema'** è stato omissa il contatore di riferimento, pertanto il conteggio dei teoremi parte da uno e prosegue di conseguenza. Si osservi invece l'esempio seguente, che presuppone un documento di tipo **'article'**:

```

\newtheorem{teorema}{Teorema}[section]

```

In questo modo, si ottiene una numerazione subordinata al contatore **'section'**, ovvero alla sequenza delle sezioni:

```

\section{Alcuni teoremi interessanti}
\begin{teorema}
In teoria, non c'è differenza tra pratica e teoria.
\end{teorema}

\begin{teorema}
In pratica, la teoria è diversa dalla pratica.
\end{teorema}

\section{Un'altra sezione}

```

```
\begin{teorema}
Un altro teorema.
\end{teorema}
```

Si può osservare che il numero mostrato dopo il titolo è strutturato, essendo composto da un prefisso che contiene il numero della sezione e da un suffisso che costituisce il numero relativo all'ambito della sezione:

1 Alcuni teoremi interessanti

Teorema 1.1 *In teoria, non c'è differenza tra pratica e teoria.*

Teorema 1.2 *In pratica, la teoria è diversa dalla pratica.*

2 Un'altra sezione

Teorema 2.1 *Un altro teorema.*

Con questa tecnica della dipendenza da un altro contatore, si possono costruire dei corollari la cui numerazione dipende dal teorema appena enunciato. Nell'esempio seguente si mostra anche la definizione degli ambienti usati:

```
\newtheorem{teorema}{Teorema}[section]
\newtheorem{corollario}{Corollario}[teorema]

\section{Alcuni teoremi e corollari interessanti}

\begin{teorema}
In teoria, non c'è differenza tra pratica e teoria.
\end{teorema}

\begin{corollario}
```

```

Bla bla bla bla.
\end{corollario}

\begin{teorema}
In pratica, la teoria è diversa dalla pratica.
\end{teorema}

\begin{corollario}
Bla bla bla bla.
\end{corollario}

```

1 Alcuni teoremi e corollari interessanti

Teorema 1.1 *In teoria, non c'è differenza tra pratica e teoria.*

Corollario 1.1.1 *Bla bla bla bla.*

Teorema 1.2 *In pratica, la teoria è diversa dalla pratica.*

Corollario 1.2.1 *Bla bla bla bla.*

Volendo invece usare una numerazione unica, sia per i teoremi, sia per i corollari, si può usare la seconda forma per la definizione dell'ambiente:

```

\newtheorem{teorema}{Teorema}[section]
\newtheorem{corollario}[teorema]{Corollario}
...

```

Ecco il risultato seguendo l'ultimo esempio già mostrato:

1 Alcuni teoremi e corollari interessanti

Teorema 1.1 *In teoria, non c'è differenza tra pratica e teoria.*

Corollario 1.2 *Bla bla bla bla.*

Teorema 1.3 *In pratica, la teoria è diversa dalla pratica.*

Corollario 1.4 *Bla bla bla bla.*

Naturalmente si possono definire anche ambienti di tipo diverso, a seconda delle esigenze; quello che conta è non usare nomi che possono creare conflitti per LaTeX.

Dal momento che queste definizioni sono numerate, diventa importante potervi fare riferimento. Per questo è necessario dichiarare un'etichetta con il comando '`\label`' all'interno dell'ambiente:

```
\newtheorem{teorema}{Teorema}[section]

\section{Alcuni teoremi e corollari interessanti}

\begin{teorema} \label{teorema-della-teoria}
In teoria, non c'è differenza tra pratica e teoria.
\end{teorema}

Purtroppo, il teorema \ref{teorema-della-teoria} viene
confutato dal teorema \ref{teorema-della-pratica}:

\begin{teorema} \label{teorema-della-pratica}
In pratica, la teoria è diversa dalla pratica.
\end{teorema}
```

1 Alcuni teoremi e corollari interessanti

Teorema 1.1 *In teoria, non c'è differenza tra pratica e teoria.*

Purtroppo, il teorema 1.1 viene confutato dal teorema 1.2:

Teorema 1.2 *In pratica, la teoria è diversa dalla pratica.*

50.14 File esterni

«

LaTeX prevede alcuni comandi per l'inclusione di file esterni; inoltre, è possibile creare file con informazioni ottenute in fase di composizione, in modo da poterle rielaborare successivamente con altri programmi.

50.14.1 Importazione

«

Esistono due comandi molto simili per dire a LaTeX di incorporare un file esterno:

Comando	Descrizione
<code>\input{<i>nome_file</i>}</code>	include il file nel punto in cui appare il comando (può essere usato anche nel preambolo);
<code>\include{<i>nome_file</i>}</code>	include il file nel punto in cui appare il comando, dopo un salto pagina (non può essere usato nel preambolo);

Comando	Descrizione
$\backslash\text{includeonly}\{\textit{nome_file_1}, \textit{nome_file_2}, \dots, \textit{nome_file_n}\}$	usato nel preambolo, consente di selezionare quali file possono essere inclusi effettivamente con il comando ‘ <code>\include</code> ’.

Se i nomi dei file non contengono l’estensione, si intende che questa sia ‘.tex’; inoltre, come si può intuire dal nome, il comando ‘`\includeonly`’ può interferire solo con le inclusioni attraverso ‘`\include`’.

50.14.2 Importazione letterale

Se si utilizza il pacchetto ‘**verbatim**’ viene messo a disposizione un comando particolare per l’inserimento di file esterni da rappresentare nella composizione in modo letterale:

Comando	Descrizione
$\backslash\text{usepackage}\{\textit{verbatim}\}$	include il pacchetto ‘ verbatim ’ (nel preambolo);
$\backslash\text{verbatiminput}\{\textit{nome_file}\}$	include il file nel punto in cui appare, rappresentandolo tale e quale.

50.14.3 Creazione di file

Per creare un file si deve prima definire il flusso in uscita, attribuendogli un nome, quindi questo deve essere aperto, inserendovi dei dati, infine lo si chiude. Di solito si creano file con una radice

uguale a quella del file contenente il sorgente, attraverso il comando ‘`\jobname`’:

Comando	Descrizione
<code>\newwrite\nome_flusso</code>	dichiara il nome del flusso da aprire in scrittura;
<code>\immediate\openout\nome_flusso</code> <code>\jobname.estensione</code>	crea il file con estensione ‘ <i>.estensione</i> ’ e lo associa al flusso indicato;
<code>\immediate\write\nome_flusso{testo}</code>	aggiunge in coda al file una riga di testo, come indicato nell’argomento;
<code>\immediate\closeout\nome_flusso</code>	chiude il file corrispondente al flusso indicato;
<code>\jobname</code>	si espande nella stringa corrispondente alla radice del nome del file LaTeX sorgente.

L’esempio seguente mostra in che modo si potrebbe realizzare un indice generale elementare:

```

...
\chapter*{Indice generale}
\input{\jobname.indice}
...
\newwrite\miofile
\immediate\openout\miofile\jobname.indice
...
\chapter*{Introduzione}
\immediate\write\miofile{Introduzione {\arabic{page}}\par}
...
\chapter{Bla bla bla}
\immediate\write\miofile{Bla bla bla {\arabic{page}}\par}

```

```

...
\appendix
\chapter{Appendice}
\immediate\write\miofile{Appendice {\arabic{page}}\par}
...
\immediate\closeout\miofile

```

All'inizio dell'esempio si vede l'inclusione di un file con lo stesso nome del sorgente LaTeX ed estensione `.indice`. Come si può intendere, la prima volta questo file non esiste e l'inclusione non produce alcun risultato. Successivamente si crea questo file, quindi, all'inizio di ogni sezione principale si annota il titolo di questa e la pagina in cui si trova. Nella composizione successiva, il file viene incluso e successivamente rigenerato, probabilmente con numeri di pagina diversi, a causa dell'allungamento del documento; nella terza composizione si dovrebbe ottenere un risultato soddisfacente. Ecco come si potrebbe presentare il file generato in questo modo:

```

Introduzione {1}\par
Bla bla bla {10}\par
Appendice {25}\par

```

Nell'esempio si deve osservare che `\arabic{page}` si è trasformato correttamente nel numero della pagina e `\par` è rimasto intatto. Nel caso si inserissero delle macro, si potrebbe ottenere un risultato indesiderabile. Per esempio, si potrebbe voler separare il titolo dal numero della pagina con dei puntini di sospensione elastici:

```

\immediate\write\miofile{Introduzione\dotfill{\arabic{page}}\par}

```

Il file generato si presenterebbe così:

```

Introduzione\unhbox \voidb@x \cleaders ←
↪\hbox to.44em{\hss .\hss }\hfill \kern \z@ {1}\par

```

In questi casi si può usare il comando ‘`\string`’ per evitare che la barra obliqua inversa del comando ‘`\dotfill`’ venga presa in considerazione:

```
\immediate\write\miofile{Introduzione\string\dotfill{\arabic{page}}\par}
```

50.15 Riferimenti

«

- *General documentation about (La)TeX*, <http://tex.loria.fr/english/general.html>
- Tobias Oetiker, *The not so short introduction to LaTeX 2e*, 2001, <http://tobi.oetiker.ch/lshort/>
- Leslie Lamport et al., *LaTeX 2e*, http://www.emerson.emory.edu/services/latex/latex2e/latex2e_toc.html, *info latex*
- Michel Goossens, Franck Mittelbach, Alexander Samarin, *The LaTeX Companion*, Addison-Wesley, 1994
- Leslie Lamport, *LaTeX, A document preparation system, User's guide and reference manual*, Addison-Wesley, seconda edizione, 1994
- Claudio Beccari, *LaTeX, Guida a un sistema di editoria elettronica*, Hoepli, 1991, ISBN 88-203-1931-4
- Robert Kiesling, *teTeX HOWTO: The Linux-teTeX Local Guide*, <http://tldp.org/HOWTO>
- Documentazione interna della distribuzione teTeX: ‘`texmf/doc/*`’, in particolare ‘`texmf/doc/latex/latex2e-html/`’, dove si trova una guida di LaTeX in HTML
- Documentazione interna Info: *info web2c*, *info tex*, *info latex*, *info pdftex*

- Sebastian Rahtz, Hàn Thé Thành, *The pdfTeX user manual*, ‘texpmf/doc/pdftex/manual.pdf’
- Vidar Bronken Gundersen, Rune Mathisen, *SGML/XML character entity reference*, <http://www.bitjungle.com/isoent/>

¹ Tutti i comandi che definiscono un livello di suddivisione del documento, da ‘**\part**’ a ‘**\subparagraph**’, consentono di utilizzare un asterisco alla fine del nome (come per esempio ‘**\section*{...}**’); l’utilizzo di questo porta a sopprimere la numerazione della voce e il suo inserimento nell’indice generale (ammesso che queste due cose siano previste normalmente).

² Eventualmente, per ottenere il carattere «tilde», oltre a usare il comando ‘**\textasciitilde**’ si può usare invece ‘**\sim**’, per produrre un carattere che a volte viene ritenuto più grazioso.

³ Naturalmente, ‘**makeindex**’ può essere usato anche specificando delle opzioni; in tal caso si può consultare la pagina di manuale *makeindex(1)*.

⁴ È bene ricordare che la tabella si compone di quattro colonne, dove la seconda e la terza non sono separate da una linea verticale.

⁵ Qui non sono state usate della parentesi quadre sintattiche per non creare confusione.

