

Una breve introduzione a L^AT_EX

Lorenzo Pantieri

24 giugno 2025

Indice

1 Storia e filosofia

2 Installare

3 Basi

4 Testo

5 Matematica

6 Tabelle e figure

7 Bibliografia

8 Indice analitico

9 Personalizzazioni

10 Revisione finale

Riferimenti bibliografici

Sommario

Questo articolo presenta L^AT_EX, un programma di composizione tipografica liberamente disponibile, indicato per scrivere documenti scientifici con la più alta qualità. Le pagine seguenti permettono di farsi un'idea sommaria di L^AT_EX e di capire se fa al proprio caso, ma non bastano per usarlo con profitto. Chi vuole farlo *deve* leggere una guida di base: la bibliografia finale ne indica alcune.

1 Storia e filosofia

Si pronuncia *l^àtek*, non *l^àtecs*!

Lorenzo Pantieri

T_EX è un software di tipocomposizione realizzato da Donald Knuth, che ha nascosto un trabocchetto nel nome del programma: T_EX, infatti, si pronuncia *tèch* (aspirando il *ch* finale) e non com'è scritto, perché è una parola scritta in greco maiuscolo.

L^AT_EX è un software realizzato da Leslie Lamport e liberamente disponibile, che usa T_EX come motore tipografico.

Per modificare un documento scritto con un comune *word processor* (come Microsoft Word), l'utente agisce direttamente sul testo come gli appare sul monitor. Con L^AT_EX, invece, l'introduzione del testo e la sua composizione avvengono *in tempi diversi*: si scrive il testo in un editor concentrandosi solo su struttura logica e contenuto del documento e lo si dà in pasto a L^AT_EX che lo impagina *successivamente*.

L^AT_EX pretende dall'utente considerazioni sul *cosa*: «il mio documento sarà formato da alcuni capitoli, ciascuno diviso in paragrafi numerati, avrà un indice, delle figure e qualche tabella». Al *come* pensa L^AT_EX.

Un file da comporre con L^AT_EX è scritto in una "lingua" costituita da istruzioni che il programma deve eseguire. Per dare l'idea di come appare un documento da comporre con L^AT_EX, si riportano alcune righe di *codice sorgente*.

```
Due matrici  $A$  e  $B$  sono \emph{simili} se  
esiste una matrice invertibile  $T$  tale che  
\begin{equation}  
 $B=T^{-1}AT$   
\end{equation}
```

Il sorgente viene composto da L^AT_EX che, attraverso T_EX, produce il documento finito. Se il risultato non soddisfa, non si può modificare direttamente il documento a schermo, ma bisogna correggere il sorgente e poi ricomporlo.

L'esempio seguente riporta a sinistra il sorgente e a destra il risultato della composizione.

```
Due matrici  $A$  e  $B$   
sono \emph{simili} se  
esiste una matrice  
invertibile  $T$   
tale che  
\begin{equation}  
 $B=T^{-1}AT$   
\end{equation}
```

Due matrici A e B sono *simili* se esiste una matrice invertibile T tale che

$$B = T^{-1}AT \quad (1)$$

Anche con pochi rudimenti di inglese si capisce quello che il linguaggio ha specificato.

2 Installare

Per creare un documento con \LaTeX sono indispensabili tre cose:

- un editor di testi con cui scrivere il file sorgente;
- il programma \LaTeX , che lo elabora e produce il documento finito;
- un programma per visualizzare il documento finito (spesso integrato nell'editor).

Ci sono molti editor che gestiscono codice \LaTeX . Si consigliano \TeX studio, multiplatforma, e \TeX Shop, specifico per Mac.

\LaTeX è uno, ma prende forma in diverse versioni che si chiamano *distribuzioni*. Una distribuzione è una raccolta di file e altro software sufficiente per produrre un documento finito.

Si consiglia di installare \TeX Live, disponibile per Windows, Mac e Linux: è una distribuzione affidabile, mantenuta da decine di sviluppatori e aggiornata annualmente.

3 Basi

Per scrivere il codice di un documento \LaTeX si crei una cartella nella quale mettere *tutti* i file del documento. Poi si apra un nuovo file e si scriva il seguente codice:

```
\documentclass[a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}

\begin{document}
Ciao, mondo!
\end{document}
```

Infine si registri il file come primo.tex (.tex è l'estensione dei sorgenti \LaTeX).

Premendo l'apposito pulsante, \LaTeX elabora il codice e produce il documento finito.

Lingue

\LaTeX è un programma nato per scrivere documenti in inglese. Per scrivere in italiano, subito dopo

la dichiarazione di classe vanno caricati altri tre *pacchetti*:

```
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}
```

Dove:

- fontenc si occupa dei font che si vedranno nel documento composto.
- T1 è la codifica dei font usati per scrivere in italiano.
- inputenc serve a \LaTeX per interpretare correttamente i caratteri immessi nell'editor.
- utf8 permette di scrivere nell'editor i caratteri accentati direttamente dalla tastiera.
- babel agisce su parole fisse, date, convenzioni tipografiche e sulla scelta delle regole di sillabazione.

File con cui si ha a che fare

Guardando nella cartella di lavoro dopo la prima composizione, si noterà che oltre al file .tex iniziale e agli eventuali file esterni come le immagini ce ne sono altri: sono i *file ausiliari* prodotti dalla composizione, e questo è un altro aspetto per cui \LaTeX è diverso dai programmi tradizionali. Il loro numero dipende dalla complessità del documento, ma è importante sapere che *non* vanno toccati: \LaTeX li crea e se ne serve automaticamente.

Comandi e ambienti

Un *comando* è un'istruzione che ordina a \LaTeX di trattare in un certo modo una porzione più o meno ampia di testo. Di regola un comando è formato da una barra rovescia \ seguita da una sequenza di caratteri alfabetici. Per esempio:

```
Gruppo Utilizzatori
Italiani di \TeX{}
e \LaTeX.
```

Gruppo Utilizzatori Italiani di \TeX e \LaTeX .

```
Sarò lì in
\emph{dieci} minuti.
```

Sarò lì in *dieci* minuti.

La giustapposizione degli elementi di un comando prende il nome di *sintassi* del comando. Ciò che va tra parentesi graffe si chiama *argomento* del comando.

Tabella 1: Caratteri speciali di \LaTeX

Carattere	Funzione
<code>\</code>	Comincia un comando
<code>{ }</code>	Delimitano un gruppo
<code>\$</code>	Delimita la matematica in linea
<code>^</code>	Esponente matematico
<code>_</code>	Pedice matematico
<code>&</code>	Separa le celle in una tabella
<code>#</code>	Numero dell'argomento
<code>~</code>	Spazio indivisibile
<code>%</code>	Commento

Un *ambiente* è una porzione di codice delimitata da un comando d'apertura e uno di chiusura. Per esempio:

```
\begin{equation*}
2+2=4
\end{equation*}
```

$$2 + 2 = 4$$

Caratteri speciali

\LaTeX interpreta in modo particolare alcuni caratteri molto richiesti nella scrittura del codice: sono i cosiddetti *caratteri speciali*, riportati nella tabella 1.

Struttura del file sorgente

\LaTeX si aspetta di trovare il sorgente da elaborare strutturato in un certo modo. Elementi fondamentali sono almeno una dichiarazione di classe

```
\documentclass{<tipo di documento>}
```

e le dichiarazioni d'inizio e fine documento:

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

Tutte le istruzioni scritte tra `\documentclass` e `\begin{document}` *inclusi* costituiscono il *preambolo del documento* (o semplicemente *preambolo*) e comprendono:

- il caricamento di pacchetti che estendono le capacità di \LaTeX ;
- le definizioni di comandi e ambienti personalizzati (si veda il paragrafo 9);
- le opzioni generali del documento.

Fra `\begin{document}` e `\end{document}` va scritto il *corpo del documento*, cioè il vero e proprio testo che

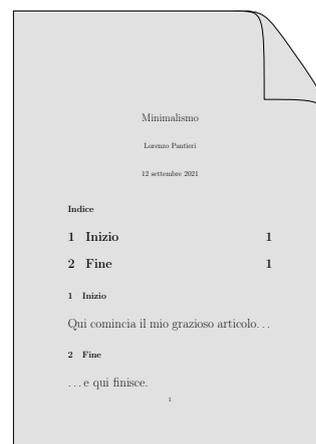


Figura 1: Documento elementare composto con \LaTeX

\LaTeX elaborerà e mostrerà nel documento finito. La figura 1 mostra il risultato della composizione del codice seguente:

```
\documentclass[a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}

\begin{document}
\author{Lorenzo Pantieri}
\title{Minimalismo}
\maketitle
\tableofcontents

\section{Inizio}
Qui comincia il mio grazioso articolo\dots

\section{Fine}
\dots e qui finisce.
\end{document}
```

Dove:

- `\begin{document}` segnala l'inizio del documento;
- `\author` e `\title` ne specificano rispettivamente nome dell'autore e titolo;
- `\maketitle` produce il contenuto dei due comandi precedenti, *dopo* i quali deve essere dato;
- `\tableofcontents` produce l'indice generale *dopo due* composizioni;
- `\section{<titolo>}` produce un titolo di sezione (un paragrafo, in questo caso);
- `\dots` produce i puntini di sospensione ... ;
- `\end{document}` segnala la fine del documento.

Spazi e righe vuote

Il modo in cui \LaTeX tratta gli spazi e le righe vuote nel sorgente è diverso da quello di tutti i comuni elaboratori di testo. Infatti:

- più spazi consecutivi sono considerati come un solo spazio, e gli spazi all'inizio di una riga vengono ignorati;
- una sola interruzione di riga è trattata come uno spazio;
- una riga vuota tra due righe di testo separa due capoversi, e più righe vuote consecutive sono trattate come una sola riga vuota.

L'esempio seguente mostra all'opera i casi appena descritti:

Non importa se si mettono uno o tanti spazi dopo una parola. E neppure se si mettono tanti spazi all'inizio di una riga o se la s'interrompe.

Le cose cambiano se si saltano una o più righe, perché in questo modo si comincia un nuovo capoverso.

Non importa se si mettono uno o tanti spazi dopo una parola. E neppure se si mettono tanti spazi all'inizio di una riga o se la s'interrompe.

Le cose cambiano se si saltano una o più righe, perché in questo modo si comincia un nuovo capoverso.

Commenti

Quando \LaTeX incontra un carattere di percento % (tranne che nella forma \%) elaborando un sorgente, ignora il resto della riga, l'interruzione di riga, e tutti gli spazi bianchi all'inizio della riga successiva. Questo carattere è utile, dunque, per appendere una nota o un promemoria (che non verrà stampato) proprio lì dove serve. Ecco all'opera:

Ecco un % semplice, % ma istruttivo esempio.

Ecco un esempio.

Classi di documento

La prima informazione che \LaTeX si aspetta di trovare nel sorgente è la *tipo di documento* che si desidera realizzare, da specificare di norma come prima cosa con il comando

Tabella 2: Comandi di sezionamento

Istruzione	Sezione
$\backslash\text{part}$	Parte
$\backslash\text{chapter}$	Capitolo
$\backslash\text{section}$	Paragrafo
$\backslash\text{subsection}$	Sottoparagrafo

$\backslash\text{documentclass}\{\langle\text{classe}\rangle\}$

dove $\langle\text{classe}\rangle$ è la classe di documento scelta.

Le principali classi di documento standard sono *article* per scrivere articoli, *report* per scrivere relazioni e *book* per scrivere libri.

Un tipico sorgente potrebbe cominciare con la riga

$\backslash\text{documentclass}[\text{a4paper},\text{twoside}]\{\text{article}\}$

che ordina a \LaTeX di impaginare il documento come un articolo, su carta di formato A4 e impostato per la stampa fronte/retro.

Strutturare il documento

Per suddividere un documento in sezioni basta dare *nel corpo del testo* i comandi elencati nella tabella 2.

Indice generale

Il comando

$\backslash\text{tableofcontents}$

produce l'indice generale. Si noti che per ottenerlo nel documento finito servono *due* composizioni successive.

Riferimenti incrociati

Nei documenti si trovano spesso riferimenti incrociati a sezioni, figure, tabelle, teoremi e altri elementi. Per realizzarli si usano i comandi $\backslash\text{label}$ e $\backslash\text{ref}$, dove:

- $\backslash\text{label}$ assegna agli elementi contrassegnati un'etichetta arbitraria e univoca;
- $\backslash\text{ref}$ produce il numero dell'elemento messo in $\langle\text{etichetta}\rangle$.

Per produrre i riferimenti incrociati nel documento sono necessarie *due* composizioni successive, altrimenti al loro posto si vedranno altrettanti ??.

Per esempio, se s'identifica un paragrafo con

```
\section{Basi}
\label{sec:basi}
```

poi ci si può riferire a esso con

```
Si veda il paragrafo
\ref{sec:basi}.
```

Si veda il paragrafo 3.

Collegamenti ipertestuali

Il pacchetto `hyperref`, che di regola va caricato *per ultimo*, crea i collegamenti ipertestuali all'interno del documento, rendendo cliccabili i riferimenti incrociati, quelli a voci bibliografiche, a indirizzi Internet e molto altro.

Pacchetti

Scrivendo un documento, prima o poi ci s'imbatta in problemi che \LaTeX non riesce a risolvere da solo. Il suo linguaggio standard, per esempio, non gestisce l'inclusione delle immagini, né permette di modificare facilmente i margini di pagina. Per aggirare ostacoli di questo tipo si sfrutta la struttura modulare del programma, che estende le proprie capacità di base tramite moduli aggiuntivi chiamati *pacchetti*.

I pacchetti si caricano *nel preambolo* con il comando

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

dove:

- *<opzioni>* è una voce o un elenco di voci separate con la virgola che specificano le impostazioni del pacchetto;
- *<pacchetto>* è il nome del pacchetto, che va scritto *sempre* in tutte minuscole.

4 Testo

Comporre i capoversi

Con l'eccezione del primo capoverso di una sezione, per cominciare un nuovo capoverso con \LaTeX basta lasciare una riga vuota nel sorgente. In ogni caso, non lo si faccia *mai* con `\`.

Un documento "ben composto" si riconosce da alcuni elementi: il testo è giustificato, le parole sono adeguatamente spaziate tra loro e sillabate a fine riga se proprio non ci stanno, i capoversi si distinguono l'uno dall'altro per facilitare la lettura. Di

Tabella 3: Comandi per modificare lo stile del font

Comando	Stile
<code>\emph</code>	<i>Evidenziato</i>
<code>\textit</code>	<i>Corsivo</i>
<code>\textsc</code>	MAIUSCOLETTO
<code>\textbf</code>	Nero
<code>\textsl</code>	<i>Inclinato</i>
<code>\textrm</code>	Tondo
<code>\textsf</code>	Senza grazie
<code>\texttt</code>	Macchina per scrivere

solito tutto questo si ottiene dando a mano le rispettive impostazioni; con \LaTeX , invece, non occorre nemmeno pensarci, perché il programma lo fa automaticamente.

Caratteri particolari e simboli

\LaTeX permette di usare accenti e caratteri particolari di molte lingue, come si può vedere nell'esempio seguente:

<code>Na\~{i}f, Stra\~{s}s e,</code>	Naïf, Straße, ¡Señorita!,
<code>!‘Se\~{n}orita!,</code>	Smørrebrød
<code>Sm\~{o} rrebr\~{o} d</code>	

Il simbolo dell'euro (€) si ottiene con il comando `\euro` del pacchetto `eurosym`.

Per altri simboli e caratteri speciali di \LaTeX , si veda una guida di base.

Modificare lo stile del font

I comandi elencati nella tabella 3 modificano lo *stile* del proprio argomento, lasciando invariato il testo successivo:

La parola che segue è in <code>\textit{corsivo}</code> .	La parola che segue è in <i>corsivo</i> . Il resto del testo è normale.
Il resto del testo è normale.	

Titoli e frontespizi

Il comando

```
\maketitle
```

dato *dopo* `\begin{document}` produce il "titolo" del documento, un blocco di informazioni definite dai comandi

```
\title{<titolo>}
\author{<autore>}
```

`\date{<data>}`

Il loro funzionamento si spiega da sé.

Il titolo generato dal comando `\maketitle` è piuttosto spartano, anche se si può accettare in articoli (come questo) e relazioni. Si consiglia di comporre il frontespizio di una tesi di laurea con il pacchetto frontespizio.

Infine, se nessuna delle soluzioni precedenti va bene, si può comporre un frontespizio personalizzato con l'ambiente `titlepage` (da aprire *subito dopo* `\begin{document}`) all'interno del quale si è completamente padroni dell'impaginazione.

Note a piè di pagina

Il comando

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

produce una nota in fondo alla pagina corrente con un riferimento nel testo costituito da un numero a esponente.¹

Le note a piè di pagina sono l'emblema della meticolosità.%
`\footnote{Ecccone una.}`

Le note a piè di pagina sono l'emblema della meticolosità.^a

^aEcccone una.

Si tenga presente che la loro numerazione riprende a ogni `\chapter` o `\section`.

Evidenziare le parole

Scrivendo a macchina, le parole importanti si evidenziano con una sottolineatura; in tipografia, invece, le parole si evidenziano mettendole *in corsivo*. Le possibilità sono due.

Per evidenziare una parola o una porzione di testo *indipendentemente dal contesto* in cui si trovano, \LaTeX definisce il comando

`\emph{<testo>}`

che si vede all'opera nell'esempio seguente:

`\emph{All'interno di un testo già evidenziato, \LaTeX{} evidenzia con lo stile tondo.}`

All'interno di un testo già evidenziato, \LaTeX evidenzia con lo stile tondo.

¹Così.

Esiste anche un altro comando, `\textit`, che produce il proprio argomento in corsivo *in ogni caso*. Per cogliere la differenza logica tra corsivo ed evidenziato, si osservino le due scritture:

`\emph{Tra un minuto.}`

Tra un minuto.

`\textit{Tra un minuto.}`

Tra un minuto.

Elenchi

In un documento gli elenchi sono molto importanti. Infatti:

- fanno “respirare” il testo;
- ne migliorano la leggibilità;
- permettono di strutturare i pensieri.

L'elenco precedente è stato ottenuto con l'ambiente `itemize` come segue:

Gli elenchi:

`\begin{itemize}`
`\item fanno “respirare” il testo;`
`\item ne migliorano la leggibilità;`
`\item permettono di strutturare i pensieri.`
`\end{itemize}`

Il comando `\item` mette un pallino nero prima di ogni elemento dell'elenco.

L'ambiente `enumerate` si usa come `itemize`, ma qui a ogni elemento `\item` premette un numero puntato:

Un elenco numerato:

`\begin{enumerate}`
`\item Mane`
`\item Tekel`
`\item Fares`
`\end{enumerate}`

Un elenco numerato:

1. Mane
2. Tekel
3. Fares

Citazioni

Esistono due modi per scrivere le citazioni con \LaTeX : “in linea” e “in display”.

Una citazione “in linea” è un testo tra virgolette appartenente al flusso del discorso, come quando si cita il motto kantiano «il cielo stellato sopra di me, la legge morale dentro di me».

Una citazione “in display” è un testo che va composto entro margini più ampi di quelli correnti e

separandolo dal contesto con adeguati spazi bianchi, in modo da metterlo “in mostra” e bene in risalto sulla pagina.

Per ottenere citazioni in display ben composte si consiglia il pacchetto quoting, da impostare come segue nel preambolo:

```
\usepackage[font=small]{quoting}
```

Il pacchetto definisce l’omonimo ambiente quoting da usare così:

```
Ecco una citazione
in display:
\begin{quoting}
Il cielo stellato
sopra di me, la legge
morale dentro di me.
\end{quoting}
La citazione è
centrata e separata
dal resto del testo.
```

Ecco una citazione in display:

```
Il cielo stellato
sopra di me, la
legge morale
dentro di me.
```

La citazione è centrata e separata dal resto del testo.

Codici

Talvolta capita di dover scrivere parole o frammenti di testo in modo *verbatim* (“alla lettera”), cioè *non* interpretando spazi, caratteri speciali, rientri, a capo, simboli e comandi. Questa modalità è utile per riportare esempi di codici informatici e linguaggi di programmazione.

Per scrivere un frammento di testo *verbatim in linea* e che non debba andare a capo, \LaTeX definisce il comando

```
\verb! (testo verbatim)!
```

Per scrivere testo *verbatim in display e su più righe*, invece, c’è l’ambiente *verbatim*, da usare come di consueto.

```
Il logo “\LaTeX”
si ottiene con il
comando
\verb!\LaTeX!.
```

Il logo “ \LaTeX ” si ottiene con il comando \LaTeX .

```
\begin{verbatim}
Nell’ambiente
verbatim i comandi
di \LaTeX,
gli
a capo, gli spazi,
i rientri e i
caratteri speciali
(\{}%$_&#^~)
non vengono
interpretati.
\end{verbatim}
```

Nell’ambiente *verbatim* i comandi di \LaTeX , gli a capo, gli spazi, i rientri e i caratteri speciali ($\{\}\%$_&#^~$) non vengono interpretati.

Questi strumenti generalmente riescono a soddisfare le esigenze più comuni, ma non si possono personalizzare in alcun modo (con colori, sfondi particolari e riquadri, per esempio). Si rimanda chi ne avesse bisogno al pacchetto listings.

5 Matematica

In questo paragrafo si danno per caricati i pacchetti *amsmath* e *amssymb*.

Formule in linea e in display

Con \LaTeX si può scrivere la matematica in due modi: “in linea” e “in display”.

Una formula “in linea” è incorporata nel testo: per esempio $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$. Come si può osservare, \LaTeX fa il possibile per comprimerla e modificare meno che può l’interlinea nel capoverso che la contiene. L’esempio seguente mostra come si scrive una formula di questo tipo:

```
Una formula in linea
è incorporata
nel testo:
$\sum_{n=1}^{\infty}
\frac{1}{n^2} =
\frac{\pi^2}{6}$.
\LaTeX{} modifica
l’interlinea il
meno possibile.
```

Una formula in linea è incorporata nel testo: $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$. \LaTeX modifica l’interlinea il meno possibile.

Le formule in linea si scrivono tra dollari $\$..\$$ e si consiglia di usarle solo con espressioni di altezza contenuta come le seguenti:

```
Ci sono voluti secoli
per dimostrare che
se $n>2$ \emph{non}
ci sono tre interi
positivi $a$, $b$,
$c$ tali che
$a^n+b^n=c^n$.
```

Ci sono voluti secoli per dimostrare che se $n > 2$ non ci sono tre interi positivi a, b, c tali che $a^n + b^n = c^n$.

Una formula “in display”, invece, è un’espressione che \LaTeX compone su linee a sé, separate dal contesto con adeguati spazi bianchi per “metterla in mostra” e farla risaltare sulla pagina. L’esempio in linea del paragrafo precedente diventa, in display:

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

Per scrivere queste formule si usa uno dei due ambienti matematici seguenti:

- `equation` per le formule numerate;
- `equation*` (di solito abbreviato in `\[...]`) per quelle non numerate.

Si considerino gli esempi seguenti:

```
Una formula in
display è composta
su linee a sé stanti:
\[
\sum_{n=1}^{\infty}
\frac{1}{n^2} =
\frac{\pi^2}{6}
\]
```

Una formula in `display` è composta su linee a sé stanti:

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

```
Se $f$ è continua e
\[
F(x)=\int_a^x f(t)\,dt
\]
allora
\begin{equation}
F'(x)=f(x)
\end{equation}
```

Se f è continua e

$$F(x) = \int_a^x f(t) dt$$

allora

$$F'(x) = f(x) \quad (2)$$

I comandi `\label` ed `\eqref` permettono i riferimenti incrociati alle formule:

```
\begin{equation}
\label{eqn:eulero}
e^{i\pi}+1=0
\end{equation}
Dalla formula
\eqref{eqn:eulero}
si deduce che\dots
```

$$e^{i\pi} + 1 = 0 \quad (3)$$

Dalla formula (3) si deduce che...

Il simbolo di sommatoria è generato da `\sum` e quello di produttoria da `\prod`. Gli estremi si scrivono come indici.

```
\[
\prod_{k=1}^n x_k
\quad
\sum_{k=1}^n x_k^2=1
\]
```

$$\prod_{k=1}^n x_k \quad \sum_{k=1}^n x_k^2 = 1$$

Una frazione si ottiene con il comando

```
\frac{\langle numeratore \rangle}{\langle denominatore \rangle}
```

Per esempio:

```
\[
\frac{1}{x^2+1}
\]
```

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

Il comando

```
\lim_{\langle variabile \rangle \to \langle valore \rangle}
```

produce i limiti, e `\infty` produce ∞ .

Tabella 4: Lettere greche

α	<code>\alpha</code>	ξ	<code>\xi</code>
β	<code>\beta</code>	Ξ	<code>\Xi</code>
γ	<code>\gamma</code>	π	<code>\pi</code>
Γ	<code>\Gamma</code>	Π	<code>\Pi</code>
δ	<code>\delta</code>	ρ	<code>\rho</code>
Δ	<code>\Delta</code>	σ	<code>\sigma</code>
ϵ	<code>\epsilon</code>	Σ	<code>\Sigma</code>
ζ	<code>\zeta</code>	τ	<code>\tau</code>
η	<code>\eta</code>	υ	<code>\upsilon</code>
ϑ	<code>\vartheta</code>	Υ	<code>\Upsilon</code>
Θ	<code>\Theta</code>	ϕ	<code>\phi</code>
ι	<code>\iota</code>	Φ	<code>\Phi</code>
κ	<code>\kappa</code>	χ	<code>\chi</code>
λ	<code>\lambda</code>	ψ	<code>\psi</code>
Λ	<code>\Lambda</code>	Ψ	<code>\Psi</code>
μ	<code>\mu</code>	ω	<code>\omega</code>
ν	<code>\nu</code>	Ω	<code>\Omega</code>

```
\[
\lim_{n \to +\infty}
f_n = \delta
\]
```

$$\lim_{n \rightarrow +\infty} f_n = \delta$$

Le derivate si scrivono con il carattere `'`, che produce il segno di *primo*.

```
\[
y=x^2 \quad \text{\quad} \quad y'=2x
\quad \quad \quad y''=2
\]
```

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

Il comando `\quad` produce lo spazio di un *quadrato*.

Il comando `\int` produce il simbolo di integrale. Gli estremi di integrazione si scrivono come indici, e un indice formato da più di una lettera o una cifra va messo tra parentesi graffe.

```
\[
\int_0^T f(x)\,dx
\]
```

$$\int_0^T f(x) dx$$

Come si può osservare, lo spazio sottile `\`, allontana dx da $f(x)$.

Le lettere greche minuscole e maiuscole si ottengono con i comandi elencati nella tabella 4.

Operatori

In \LaTeX , le funzioni come `\sin`, `\cos` e `\log` sono prodotte dai comandi `\sin`, `\cos` e `\log`, detti *operatori*. Per esempio:

```
\[
\sin2x \quad \log(xy)
\]
```

$$\sin 2x \quad \log(xy)$$

Parentesi

\LaTeX definisce molti simboli per parentesi e altri delimitatori. Le parentesi tonde e quadre si scrivono con i corrispondenti caratteri da tastiera, mentre quelle graffe devono essere precedute da `\`. Tutti gli altri delimitatori vengono generati da comandi dedicati.

```
\[
\{a,b,c\}\ne\{a,b,c\}
\]
```

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

Vettori e matrici

I vettori si scrivono di solito in neretto; talvolta, soprattutto nei testi di fisica, sono sormontati da una freccia. Nel primo caso si può usare il comando `\boldsymbol`; nel secondo il comando `\vec`. Può essere conveniente ridefinire nel preambolo quest'ultimo comando (si veda il paragrafo 9):

```
\renewcommand{\vec}{\boldsymbol}
```

In questo modo basta scrivere `\vec{v}` per ottenere v e si può cambiare notazione con un'unica modifica.

Le matrici si scrivono negli ambienti `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` e `Vmatrix`, che hanno come delimitatori rispettivamente parentesi tonde, quadre, graffe, barre verticali e doppie barre verticali. Esiste anche l'ambiente `matrix` senza delimitatori.

```
\[
\begin{pmatrix}
1 & 2 \\
3 & 4
\end{pmatrix}
\end{pmatrix}
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Enunciati e dimostrazioni

Di qui fino alla fine del paragrafo si dà per caricato *anche* il pacchetto `amsthm`.

Nella scrittura della matematica è utile poter disporre di un metodo per introdurre e numerare definizioni, teoremi e strutture simili. I tipi di enunciato non sono predefiniti, ma vanno dichiarati dall'utente, chiamato a prendere alcune decisioni globali:

- stabilire il tipo di enunciato da inserire (per esempio, definizioni e teoremi);
- assegnare un nome a ogni ambiente (per esempio, definizione e teorema)
- titolare gli enunciati (per esempio, con *Definizione* e *Teorema*).

Il comando `\newtheorem`, dato *nel preambolo*, permette di fare le relative dichiarazioni globali:

```
\newtheorem{nome dell'enunciato}{titolo}
```

dove:

- `\langle nome dell'enunciato \rangle` è una parola chiave che identifica l'enunciato;
- `\langle titolo \rangle` specifica il titolo dell'enunciato che comparirà nel documento.

Il testo dell'enunciato va messo nel corrispondente ambiente, così:

```
\begin{nome dell'enunciato}
...
\end{nome dell'enunciato}
```

Il pacchetto `amsthm` fornisce due stili predefiniti per gli enunciati: `plain`, per i teoremi, e `definition`, per le definizioni. Il primo produce il proprio contenuto in corsivo, mentre il secondo lo lascia in tondo.

L'ambiente `proof` permette di scrivere una dimostrazione, che nel documento finito sarà chiusa da un quadratino.

A questo punto le nozioni teoriche dovrebbero bastare. Alcuni esempi mostreranno quanto si è appena esaminato. Scrivendo nel preambolo

```
\theoremstyle{definition}
\newtheorem{definizione}{Definizione}
```

e

```
\theoremstyle{plain}
\newtheorem{teorema}{Teorema}
```

gli ambienti `definizione` e `teorema` si usano così:

```
\begin{definizione}
Un numero naturale è
\emph{primo} se è
divisibile solo per
sé stesso e per $1$.
\end{definizione}
```

Definizione 1. Un numero naturale è *primo* se è divisibile solo per sé stesso e per 1.

```
\begin{teorema}
I numeri primi
sono infiniti.
\end{teorema}
```

Teorema 1. *I numeri primi sono infiniti.*

```
\begin{proof}
Per esercizio.
\end{proof}
```

```
\begin{teorema}
In un triangolo
rettangolo, la somma
dei quadrati
costruiti sui cateti
è uguale al quadrato
costruito
sull'ipotenusa.
\end{teorema}
```

Dimostrazione. Per esercizio. \square

Teorema 2. *In un triangolo rettangolo, la somma dei quadrati costruiti sui cateti è uguale al quadrato costruito sull'ipotenusa.*

La figura

```
\begin{center}
\includegraphics[%
width=\textwidth]{%
Rettili}
\end{center}
```

riproduce l'incisione `\emph{Tassellazione del piano con rettili}` di Maurits Escher.

La figura



riproduce l'incisione *Tassellazione del piano con rettili* di Maurits Escher.

Come si può osservare, \LaTeX :

- numera automaticamente gli enunciati;
- mette le definizioni in tondo e i teoremi in corsivo.

6 Tabelle e figure

Tabelle e figure sono tra gli “oggetti” più problematici, perché non si possono spezzare su più pagine. In questo paragrafo, che spiega come servirsene senza sorprese, si danno per caricati i pacchetti `booktabs` e `caption` per le tabelle, e `graphicx` per le figure.

Strumenti fondamentali

Per inserire tabelle e figure in un documento da comporre con \LaTeX esistono due strumenti essenzialmente:

- l'ambiente `tabular`, per le tabelle;
- il comando `\includegraphics` definito dal pacchetto `graphicx`, per includere nel documento le figure.

Li si vede all'opera nei due esempi seguenti.

La tabella

```
\begin{center}
\begin{tabular}{ll}
\toprule
 $f(x)$  &  $f'(x)$  \\
\midrule
 $x^2$  &  $2x$  \\
 $e^x$  &  $e^x$  \\
\bottomrule
\end{tabular}
\end{center}
```

mostra le derivate di due funzioni.

La tabella

$f(x)$	$f'(x)$
x^2	$2x$
e^x	e^x

mostra le derivate di due funzioni.

Si noti che:

- Tutti e due *non* cominciano un nuovo capoverso, ma producono un'unità tipografica indivisibile che il programma tratta come se fosse un unico carattere, e che va centrata rispetto alla giustezza del testo mettendola nell'ambiente `center`.
- Quando richiesta, si consiglia di assegnare all'oggetto una larghezza *relativa* espressa con una frazione della giustezza stabilita dalla classe in uso (`\textwidth`). Una larghezza *assoluta* causerebbe inconvenienti cambiando classe di documento o aumentando le colonne di composizione.

Oggetti in testo e fuori testo

In tipografia esistono due tipi di oggetto: “in testo” e “fuori testo”.

Osservando gli esempi del paragrafo precedente, si possono notare le caratteristiche degli oggetti “in testo”, i quali:

- appartengono al flusso del discorso e non possono esserne scorporati senza comprometterne la comprensione;
- non prevedono didascalia né riferimenti incrociati a sé stessi.

Apparentemente innocui, oggetti di questo tipo possono comportare in realtà problemi di impaginazione a volte irrisolvibili. S'immagini, per esempio, di essere arrivati quasi alla fine della pagina, e di dover inserire *proprio lì*, perché richiesto dal discorso, una figura alta cinque centimetri avendone però soltanto tre a disposizione: va da sé che *lì* la figura *non* ci può stare in nessun modo: se lo spazio fisico non c'è, non lo si può inventare!

Ecco perché gli oggetti in testo devono essere *eccezionali* (tornano utili per mettere un logo “proprio

li" e in pochissime altre circostanze) e di piccole dimensioni.

Si risolve il problema rendendo gli oggetti "fuori testo" (o "mobili"; in inglese *floating*, "galleggianti") e lasciando fare a L^AT_EX che, nell'ordine in cui oggetti dello stesso tipo sono definiti nel sorgente, li metterà nel punto *per lui* migliore (sulla pagina corrente se ci stanno, oppure in pagine *successive* a quella in cui finirebbero) riempiendo lo spazio rimanente con l'altro materiale a disposizione.

A differenza di quelli in testo, gli oggetti fuori testo:

- non appartengono al flusso del discorso e per esigenze tipografiche possono essere spostati altrove dal punto esatto in cui stanno nel sorgente;
- devono avere *obbligatoriamente* un'etichetta, un numero progressivo per gli eventuali riferimenti incrociati e una didascalia che ne descriva il contenuto.

Indipendentemente dalle dimensioni degli oggetti, perciò, si raccomanda di includerli *sempre* così, vincendo le iniziali perplessità derivanti dal vederli molto spesso in un punto diverso da quello in cui li si è definiti.

Per rendere mobile un oggetto basta inserirne il relativo codice nell'ambiente `table`

```
\begin{table}
...
\end{table}
```

se è una tabella, oppure in quello `figure`

```
\begin{figure}
...
\end{figure}
```

se è una figura.

Il comando

```
\caption{<didascalia>}
```

produce, nell'ordine, l'intestazione *Tabella* o *Figura*, il numero progressivo dell'oggetto e la sua *<didascalia>*.

Il comando `\label`, da dare sempre *dopo* il corrispondente `\caption`, assegna all'oggetto un'etichetta per i riferimenti incrociati.

Il modo migliore per introdurre un oggetto mobile nel sorgente è scriverne il relativo ambiente preceduto e seguito da una riga vuota. Ecco un esem-

pio tipico per una tabella `tabular` con il relativo richiamo:

```
\dots qui finisce un capoverso.
```

```
\begin{table}
\caption{<...>}
\label{tab:esempio}
\centering
\begin{tabular}{<...>}
...
\end{tabular}
\end{table}
```

La tabella~\ref{tab:esempio} è mobile.

E uno per una figura:

```
\dots qui finisce un capoverso.
```

```
\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{<...>}
\caption{<...>}
\label{fig:esempio}
\end{figure}
```

La figura~\ref{fig:esempio} è mobile.

Come si può osservare:

- per centrare un oggetto mobile sulla pagina si usa `\centering`, perché l'ambiente `center` lascia tra testo e oggetto uno spazio verticale eccessivo (ma adeguato per un oggetto in testo);
- la corretta posizione della didascalia varia a seconda dell'oggetto cui è apposta: la tradizione italiana la vuole prima di una tabella e dopo una figura, e in queste posizioni *deve* essere messa anche nel sorgente.

Durante la stesura del documento bisogna concentrarsi sul contenuto del lavoro, lasciando fare a L^AT_EX tutto il resto. La gestione degli oggetti non si sottrae a questa regola. In linea di massima si troveranno gli oggetti "abbastanza" vicini al punto in cui stanno nel sorgente.

Non ci si lamenti per non vederli esattamente dove li si è inseriti nel sorgente! Non è un difetto di L^AT_EX, questo, ma un vantaggio, proprio come lo è il non doversi preoccupare di numerare a mano sezioni e pagine o il non dover pensare a quanto spazio ci va tra un titolo e il testo successivo.

Nel paragrafo 10 si mostra che cosa (eventualmente) si può fare durante la revisione per risolvere col-

Tabella 5: Descrittori standard delle colonne

Descrittore	Spiegazione
l	Allinea il contenuto della cella a sinistra
c	Centra il contenuto della cella
r	Allinea il contenuto della cella a destra
p	Giustifica un testo lungo entro una $\langle larghezza \rangle$

locazioni poco gradite e migliorare ulteriormente il proprio documento.

Table

Le celle di una tabella vanno separate tra loro con il carattere separatore `&` e le righe *devono* terminare con il comando `\`, pena un errore.

I filetti sono prodotti da tre comandi definiti dal pacchetto `booktabs`. Questi comandi non vogliono `\` dopo di sé e vanno dati secondo un ordine rigoroso:

1. `\toprule` produce il primo filetto della tabella;
2. `\midrule` produce il filetto interno;
3. `\bottomrule` produce l'ultimo filetto.

Anche se \LaTeX non richiede di incolonnare le celle nel sorgente, si consiglia di farlo ugualmente: un codice ordinato facilita eventuali modifiche, diminuisce la probabilità di commettere errori e aumenta quella di scovarli.

Si osservi che:

- `tabular` richiede un argomento formato da un certo numero di *descrittori*, ciascuno dei quali definisce il comportamento di un *tipo di colonna* come spiegato nella tabella 5;
- eventuali formule matematiche si scrivono con i comandi per le formule in linea, per esempio fra dollari $\$. . \$$.

Le tabelle migliori si ottengono lasciando loro la propria larghezza naturale. È ciò che fanno i tre descrittori `l c r`, allargando automaticamente la cella in base al contenuto. Se quest'ultimo è costituito da un testo troppo lungo, però, la tabella eccede la giustezza della riga e \LaTeX lo notifica con il relativo avviso. Per queste celle non si possono più

Tabella 6: Tabella con colonna p

Forza	Una forza è una grandezza fisica che cambia lo stato di quiete o di moto dei corpi.
Momento	Il momento di una forza rispetto a un punto è il prodotto vettoriale tra la posizione e la forza.

usare le colonne appena viste, ma bisogna ricorrere ad altri strumenti, per esempio a un descrittore `p{\langle larghezza \rangle}`, che permette di stabilire a priori la larghezza di una sola colonna.

Il codice seguente, che mostra all'opera il descrittore `p{\langle larghezza \rangle}`, produce la tabella 6:

```
\begin{tabular}{lp{0.5\textwidth}}
\toprule
\textbf{Forza} & Una forza è una grandezza
fisica che cambia lo stato di quiete o di
moto dei corpi. \\
\midrule
\textbf{Momento} & Il momento di una forza
rispetto a un punto è il prodotto vettoriale
tra la posizione e la forza. \\
\bottomrule
\end{tabular}
```

Si noti che:

- `\textbf` produce il proprio argomento in nero;
- per impostazione predefinita, il contenuto di una colonna `p` viene giustificato e sillabato automaticamente;
- le eventuali colonne `l`, `c` e `r` rimangono della propria larghezza naturale.

Figure

Si possono dividere le figure in due grandi classi: le immagini *vettoriali* e le immagini *bitmap*.

Le immagini vettoriali sono descritte da forme, possono essere scalate e deformate senza perdere in definizione e sono adatte soprattutto per schemi e grafici. Le si può aggiungere al documento dopo averle preparate a parte con programmi specifici. Il formato vettoriale più noto e diffuso è il PDF.

Le immagini bitmap sono matrici di pixel colorati, di solito perdono in definizione se ingrandite o

rimpicciolite e sono più adatte a fotografie e icone. I formati bitmap più diffusi sono il JPEG e il PNG.

Il pacchetto `graphicx` gestisce il trattamento delle immagini. Il comando `\includegraphics`, la cui sintassi è

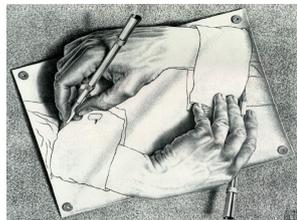
```
\includegraphics[<chiave>=<valore>]{<immagine>}
```

le include nel documento. Si osservi che:

- nell'argomento facoltativo ci vanno le opzioni che regolano l'aspetto della figura sulla pagina nella forma *<chiave>=<valore>*;
- nell'argomento obbligatorio ci va il nome dell'immagine *senza* specificarne l'estensione.

L'utente può assegnare all'immagine una larghezza (`width`) a piacere

```
\includegraphics[width=\textwidth]{%mani}
```



7 Bibliografia

La bibliografia è uno degli aspetti più delicati di un documento, e \LaTeX aiuta anche in questo caso, definendo gli strumenti per realizzarla e gestirla con semplicità.

L'ambiente `thebibliography` gestisce la bibliografia di un documento molto facilmente. La sintassi generale è:

```
\begin{thebibliography}{<etichetta più lunga>}
\bibitem{<chiave di citazione>}
...
\end{thebibliography}
```

dove:

- *<etichetta più lunga>* può essere un numero (di solito 9 se la bibliografia comprende meno di dieci opere, 99 se almeno dieci ma meno di cento, eccetera) oppure del testo (nel caso di etichette personalizzate: si scrive allora l'etichetta più lunga);
- `\bibitem` va premesso a ogni riferimento bibliografico;
- *<chiave di citazione>* è l'etichetta *univoca* per citare la fonte nel documento (si consiglia di usare la sintassi *<autore>: <titolo>* analoga a quella di `\label`).

Lo si vede all'opera nell'esempio seguente:

```
\begin{%
thebibliography}{9}
\bibitem{eco:tesi}
Eco, Umberto (1977),
\emph{Come si fa
una tesi di laurea},
Bompiani, Milano.

\bibitem{mori:tesi}
Mori, Lapo (2007),
“Scrivere con  $\LaTeX$ ”,
ArsTeXnica.
\end{thebibliography}
```

Bibliografia

- [1] Eco, Umberto (1977), *Come si fa una tesi di laurea*, Bompiani, Milano.
- [2] Mori, Lapo (2007), “Scrivere con \LaTeX ”, ArsTeXnica.

Si noti che:

- `thebibliography` si comporta in modo molto simile a un ambiente per elenchi;
- ciascun riferimento bibliografico va scritto per intero, regolandone *a mano* tutti gli aspetti (corsivo, virgolette, eccetera), compresa la posizione nell'ordine alfabetico;
- nel documento finito i riferimenti saranno contrassegnati con un numero tra parentesi quadre sia nella bibliografia sia nelle citazioni.

Per citare un riferimento bibliografico nel testo si usa il comando `\cite`. Il prossimo esempio lo mostra all'opera:

```
Vedi \cite{eco:tesi}
per i dettagli.
```

Vedi [1] per i dettagli.

8 Indice analitico

L'indice analitico è un elenco alfabetico di parole o espressioni dette *voci*, posto di regola alla fine di un documento; accanto a ogni voce ci sono i numeri delle pagine in cui la voce in questione compare. In molti lavori questo indice è utilissimo, e \LaTeX è capace di gestirlo automaticamente e con grande efficienza, come si mostra in questo capitolo.

Per creare l'indice analitico con \LaTeX bisogna innanzitutto eseguire due operazioni preliminari:

1. caricare il pacchetto `imakeidx` per abilitare il programma alla composizione dell'indice;
2. dare *nel preambolo* il comando `\makeindex`, per attivare i comandi dedicati che verranno inseriti nel corpo del testo.

A questo punto, *immediatamente dopo* ogni parola o espressione da indicizzare basterà dare il comando

```
\index{<voce>}
```

Per produrre l'indice analitico basta dare il comando `\printindex` immediatamente prima di `\end{document}` e compilare con \LaTeX .

Per esempio, il codice:

```
\documentclass{article}
\usepackage{imakeidx}
\makeindex
```

```
\begin{document}
```

Si possono inserire nell'indice analitico parole come `\emph{arte}``\index{arte}` o `interesse``\index{interesse}`.

```
\printindex
\end{document}
```

crea un indice analitico con le voci *arte* e *interesse*.

9 Personalizzazioni

S'immagini di scrivere un libro di botanica e di volere tutti i nomi di pianta in corsivo. Il modo più veloce per farlo è scrivere ogni nome nell'argomento di `\textit`. Successivamente l'editore chiede i nomi di pianta in nero. Che fare? Si potrebbero sostituire automaticamente tutti i `\textit` con altrettanti `\textbf`, ma ci sarebbe qualche problema, perché si potrebbe aver usato il corsivo anche in altre parti del libro, che invece devono rimanere tali. Non resta che sostituire un comando dopo l'altro perdendo un sacco di tempo.

Con \LaTeX , anziché agire a mano si può definire *una volta per tutte* un nuovo comando, `\pianta`, che produce il proprio argomento nello stile deciso dall'utente. Se l'imposizione del nero avviene a documento completato, basterà modificare *una volta per tutte* la definizione del comando.

I comandi personali si definiscono *nel preambolo* con `\newcommand`:

```
\newcommand{<nome>}{<definizione>}
```

Dove:

- `<nome>` è il nome che si dà al nuovo comando.
- `<definizione>` sono le istruzioni che specificano ciò che si vuole che il nuovo comando "faccia".

Di seguito si mostra la sintassi di un comando personale:

```
\newcommand{\pianta}{\textit}
```

Quando si dà il comando, il testo nell'argomento di `\pianta` viene trattato secondo la *<definizione>*. In questo caso verrà reso in corsivo, come si vede nell'esempio seguente:

```
\pianta{Rosa canina}      Rosa canina
```

Un ambiente personale *senza argomenti* si definisce con il comando `\newenvironment` *nel preambolo*:

```
\newenvironment{<nome>}%
  {<comandi di apertura>}{<comandi di chiusura>}
```

la cui sintassi si spiega da sé.

L'esempio seguente mostra la sintassi di un ambiente personale. Scrivendo nel preambolo

```
\newenvironment{itaitemize}%
  {\begin{itemize}\itshape}{\end{itemize}}
```

si potrà usare il nuovo ambiente `itaitemize` così:

```
\begin{itaitemize}
\item un elenco                      • un elenco
\item con voci                        • con voci
\item corsive                         • corsive
\end{itaitemize}
```

10 Revisione finale

Anche se \LaTeX induce a privilegiare la struttura logica del documento e a trascurarne l'aspetto finale, nemmeno lui risolve *tutti* i problemi tipografici evitando d'intervenire a mano. La revisione finale del documento è un'arte complicata ma ricca di soddisfazioni, che dà i propri frutti migliori solo quando il documento è nella sua forma *definitiva*.

I difetti d'impaginazione orizzontali riguardano la formazione dei capoversi (di solito consistono in righe sporgenti nel margine destro) e dei titoli di sezione, che \LaTeX potrebbe spezzare in modo insoddisfacente. Si risolvono tutti i problemi riformulandoli: si può fare praticamente sempre.

I difetti d'impaginazione verticali riguardano la divisione in pagine del documento.

Gli oggetti in testo causano problemi d'impaginazione a volte irrisolvibili perché, di regola, non possono essere spezzati tra due pagine. Basta evitarli il più possibile.

Gli oggetti fuori testo, invece, sono molto più flessibili. Tenendo presente che si può considerare “ottimale” il risultato fintanto che oggetto e relativo riferimento si trovano sulla stessa pagina o al massimo in due pagine opposte, qualche oggetto potrebbe comunque non piacere dove \LaTeX ha pensato di metterlo. Questi casi si risolvono spesso spostando il codice dell’oggetto di qualche capoverso.

In tipografia si usa chiamare *orfano* la prima e unica riga di un capoverso in fondo alla pagina e *vedova* l’ultima riga di un capoverso in cima a una pagina nuova. Sono orrori tipografici da evitare con cura certissima. \LaTeX è programmato per evitare automaticamente queste due situazioni: quando non ce la fa, per risolvere la faccenda basta riformulare il capoverso in questione per diminuirlo o aumentarlo di una riga oppure valutare la possibilità di spezzare un capoverso in due o di riunirne due in uno nei pressi del problema.

Infine, non rimane che rileggere il tutto *più volte* per stanare i refusi: è praticamente impossibile non scovarne in ogni documento che superi la decina di pagine.

Riferimenti bibliografici

Beccari, Claudio

2021 *Introduzione all’arte della composizione tipografica con \LaTeX* , <http://www.guitex.org/home/images/doc/guidaguit-b5.pdf>.

Oetiker, Tobias, Hubert Partl, Irene Hyna e Elisabeth Schlegl

2020 *Una Una mica tanto breve introduzione a $\LaTeX 2\epsilon$* , <https://ctan.org/tex-archive/info/lshort/italian/it-lshort.pdf>.

Pantieri, Lorenzo

2021a *\LaTeX pedia*, http://www.lorenzopantieri.net/LaTeX_files/LaTeXpedia.pdf.

2021b *\LaTeX per l’impaziente*, http://www.lorenzopantieri.net/LaTeX_files/LaTeXImpaziente.pdf.

Pantieri, Lorenzo e Tommaso Gordini

2021 *L’arte di scrivere con \LaTeX* , http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf.