

Lorenzo Pantieri
 $\LaTeX$ per l'impaziente



Un'introduzione all'Arte di scrivere con $\LaTeX$

Gruppo Utilizzatori Italiani di $\TeX$ e $\LaTeX$
<http://www.guit.sssup.it/>
Ottobre 2009



Lorenzo Pantieri: *LaTeX per l'impaziente*, Un'introduzione all'Arte di scrivere con LaTeX, © ottobre 2009.

Sito Web:

<http://www.lorenzopantieri.net/>

E-mail:

lorenzo.pantieri@iperbole.bologna.it

Nel frontespizio sono riprodotte una litografia e un'incisione di Maurits C. Escher, dal titolo *Belvedere* e *Tassellazione del piano con uccelli* (le immagini sono tratte da <http://www.mcescher.com/>). L'immagine riprodotta in questa pagina ritrae "il leone TeX", disegnato da Duane Bibby (<http://www.ctan.org/lion.html>).

Sommario

L^AT_EX è un programma di composizione tipografica liberamente disponibile, particolarmente indicato per l'elaborazione di documenti scientifici, ai più elevati livelli di qualità. Lo scopo di questo lavoro, basato sulla mia guida “L’arte di scrivere con L^AT_EX” (cui si rimanda per ogni approfondimento) e rivolto sia ai principianti di L^AT_EX sia a coloro che già lo conoscono, è fornire agli utenti di L^AT_EX di lingua italiana gli elementi essenziali per comporre un documento usando questo potentissimo strumento di scrittura.

Tale obiettivo è perseguito innanzitutto presentando le nozioni fondamentali del programma, nella maniera più chiara e sintetica possibile. Al contempo, vengono forniti svariati esempi e vengono analizzati alcuni problemi tipici incontrati durante la stesura di una pubblicazione scientifica o professionale, specialmente in lingua italiana, indicando le soluzioni che ritengo migliori.

La scelta delle soluzioni adottate deriva principalmente dalle discussioni presenti sul forum del Gruppo Utilizzatori Italiani di T_EX e L^AT_EX (<http://www.guit.sssup.it/>), che resta un eccellente riferimento per i temi trattati in questo documento.

Abstract

L^AT_EX is a free typesetting system, particularly useful to elaborate scientific documents, at the highest standards of quality. The purpose of this work, based on my guide “The art of writing with L^AT_EX” (in Italian, to which I refer the reader for any deepening) and devoted both to the beginners of L^AT_EX and to those who already know it, is to provide Italian L^AT_EX users the essential tools to create a document using this powerful writing tool.

This aim is pursued by introducing the fundamental notions of the program, in the most synthetic and clearest way. At the same time, this work shows many examples and analyzes the typical problems faced during the writing of an academic or professional publication, especially in Italian, indicating the solutions I consider best.

The choice of the solutions mainly stems from the several topics on the forum of the Italian T_EX User Group (<http://www.guit.sssup.it/>), which is always an excellent reference for all the themes dealt with in this document.

Ringraziamenti

Desidero ringraziare in primo luogo i membri dello Staff del Gruppo Utilizzatori Italiani di $\text{\TeX}$ e $\text{\LaTeX}$ ($\text{\GjT}$, <http://www.guit.sssup.it/>), in particolare il Prof. Claudio Beccari, Fabiano Busdraghi, Gustavo Cevolani, Massimiliano Domini, il Prof. Enrico Gregorio, Lapo F. Mori, Andrea Tonelli, Emiliano G. Vavassori ed Emanuele Vicentini, per l'impagabile aiuto fornito nella redazione di questo lavoro, le spiegazioni dettagliate, la pazienza e la precisione nei suggerimenti, le soluzioni fornite, la competenza e la disponibilità: grazie mille, ragazzi! Grazie anche tutti quelli che hanno discusso con me sul forum del $\text{\GjT}$, prodighi di preziose osservazioni e di validi consigli.

Rivolgo un ringraziamento davvero particolare al Prof. Enrico Gregorio e al Prof. Claudio Beccari. La loro alta professionalità ed esperienza, unite alla gentilezza e disponibilità, sono state indispensabili per la riuscita di questo studio.

E un grazie davvero speciale a Monia, che mi è stata vicina in questo periodo intenso, piacevole e costruttivo.

Cesena, 1 ottobre 2009

L. P.

Indice

1	Introduzione	1
2	Storia e filosofia	3
2.1	Storia	3
2.1.1	TeX	3
2.1.2	Etimologia	3
2.1.3	L ^A TeX	4
2.2	Filosofia	4
2.2.1	Composizione sincrona e asincrona	4
2.2.2	Concentrarsi sul contenuto e non sulla forma	5
3	Installare L^ATeX	7
3.1	Introduzione	7
3.2	Installazione per Windows	8
3.2.1	La distribuzione MiKTeX	8
3.2.2	La distribuzione TeX Live per Windows	9
3.2.3	Editor per Windows	9
3.3	Installazione per Mac	11
3.3.1	La distribuzione MacTeX	11
3.3.2	Editor per Mac	11
3.4	Installazione per Linux	11
3.4.1	La distribuzione TeX Live per Linux	12
3.4.2	Editor per Linux	12
3.5	Altri programmi utili	12
3.6	L ^A TeX e PDFL ^A TeX	13
4	Le basi	15
4.1	Il nostro primo documento con L ^A TeX	15
4.1.1	La scrittura del sorgente	15
4.1.2	La compilazione	16
4.1.3	La correzione degli errori	16
4.1.4	La visualizzazione	17
4.1.5	La stampa	17
4.2	I file sorgenti di L ^A TeX	17
4.2.1	Spazi	17
4.2.2	I caratteri speciali	18
4.2.3	I comandi	20

4.2.4	I commenti	20
4.3	La struttura del file sorgente	21
4.4	Le classi di documento	22
4.5	I pacchetti	24
4.5.1	Che cosa sono?	24
4.5.2	Come sapere se se ne ha bisogno?	24
4.5.3	Come scoprire qual è il pacchetto “giusto”?	25
4.5.4	Come installarli?	25
4.5.5	I pacchetti di uso più comune	28
4.6	Gli stili di pagina	29
4.7	I file con cui si ha a che fare	30
4.7.1	I file dell’utente	30
4.7.2	File di classi, pacchetti e stili	30
4.7.3	I file ausiliari	30
4.7.4	I file di output	31
4.8	Documenti di grandi dimensioni	31
5	Il testo	33
5.1	LaTeX multilingue e multiplatforma	33
5.1.1	Il pacchetto babel	33
5.1.2	Il pacchetto inputenc	35
5.2	La codifica dei font	35
5.3	La struttura del testo	36
5.4	La composizione dei capoversi	37
5.4.1	La divisione delle parole in fin di riga	39
5.4.2	Lo spazio tra le parole	40
5.4.3	Il pacchetto microtype	41
5.4.4	Il rientro sulla prima riga	41
5.5	Le proporzioni di pagina	42
5.5.1	Il tormentone dei margini	42
5.5.2	L’interlinea	43
5.6	Il sezionamento del documento	43
5.6.1	Materiale iniziale, principale e finale	44
5.6.2	L’indice generale	44
5.7	Caratteri speciali e simboli	45
5.7.1	Virgolette, tratti e punti ellittici	45
5.7.2	Indirizzi Internet e riferimenti ipertestuali	46
5.7.3	Loghi, accenti e caratteri speciali	47
5.8	Il titolo del documento e il frontespizio	48
5.9	I riferimenti incrociati	49
5.10	Note a margine e a piè di pagina	50
5.11	Parole evidenziate	51
5.12	Ambienti	51
5.12.1	Elenchi puntati, numerati e descrizioni	51
5.12.2	Centrare e allineare i capoversi	53
5.12.3	Citazioni e versi	54
5.12.4	Codici	55

5.13	La revisione finale	56
6	Tabelle e figure	59
6.1	Le tabelle	59
6.1.1	Regole generali	59
6.1.2	L'ambiente tabular	59
6.1.3	Celle su più colonne	61
6.1.4	Il pacchetto tabularx	61
6.1.5	Allineare i numeri alla virgola	62
6.1.6	Le tabelle mobili	62
6.2	Le figure	64
6.2.1	Immagini vettoriali e bitmap	64
6.2.2	Conversione dei formati	65
6.2.3	Scontornare le immagini	66
6.2.4	L'inclusione delle immagini	67
6.3	Figure e tabelle in testo e fuori testo	68
6.4	Gli oggetti mobili	70
6.4.1	Gli ambienti table e figure	71
6.4.2	Che cosa fare durante la stesura del testo	73
6.4.3	Che cosa fare durante la revisione del testo	74
6.4.4	Personalizzare le didascalie: il pacchetto caption	75
6.4.5	Affiancare figure o tabelle: il pacchetto subfig	75
7	La matematica	77
7.1	Formule in corpo e fuori corpo	77
7.2	Nozioni basilari	79
7.2.1	Raggruppamenti	79
7.2.2	Apici, pedici e radici	79
7.2.3	Somme, prodotti e frazioni	80
7.2.4	Limiti, derivate e integrali	80
7.2.5	Insiemi numerici	81
7.2.6	Lettere greche	81
7.2.7	Accostare simboli ad altri simboli	82
7.2.8	Barre e accenti	82
7.2.9	Punti e frecce	83
7.2.10	Spazi in modo matematico	84
7.3	Gli operatori	84
7.4	Le parentesi	86
7.5	Vettori e matrici	88
7.6	Formule fuori corpo	89
7.6.1	Formule spezzate senza allineamento: multline	90
7.6.2	Formule spezzate con allineamento: split	90
7.6.3	Gruppi di formule senza allineamento: gather	90
7.6.4	Gruppi di formule con allineamento: align	91
7.6.5	Gli ambienti gathered e aligned	91
7.6.6	Casi e numerazione subordinata	92
7.7	Modificare lo stile e il corpo dei font	92

7.8	Enunciati e dimostrazioni	94
7.9	Diagrammi commutativi	97
7.10	Elenco dei simboli matematici	98
8	La bibliografia	105
8.1	L'ambiente thebibliography	105
9	L'indice analitico	107
9.1	Creare l'indice analitico	107
9.2	Bilanciare le colonne dell'indice analitico	109
10	Personalizzare L^AT_EX	111
10.1	Comandi, ambienti e pacchetti nuovi	111
10.1.1	Definire nuovi comandi	111
10.1.2	Nuovi ambienti	112
10.1.3	Un pacchetto personale	113
10.2	Font	113
10.2.1	Comandi per cambiare lo stile dei font	113
10.2.2	Dichiarazioni per cambiare il corpo dei font	114
10.3	Le pagine bianche nei documenti fronte-retro	115
A	Norme tipografiche italiane	117
A.1	L'accento e l'apostrofo	117
A.1.1	Uso dell'accento	117
A.1.2	Apostrofo	118
A.2	Punteggiatura e spaziatura	118
A.2.1	Segni di interpunzione	118
A.2.2	Punti ellittici	119
A.3	Stile dei font	119
A.3.1	Corsivo	119
A.3.2	Neretto	119
A.4	Capoversi	119
	Bibliografia	121
	Indice analitico	123

Elenco delle figure

3.1	Alcuni editor per $\text{\LaTeX}$	10
4.1	Un breve documento	22
4.2	La gestione automatica dei pacchetti	26
5.1	Un esempio d'uso del pacchetto frontespizio	49
6.1	Una figura posizionata manualmente.	75
6.2	Un esempio d'uso del pacchetto subfig	76

Elenco delle tabelle

4.1	Alcuni codici ASCII	19
4.2	Opzioni delle classi standard	24
5.1	Le principali codifiche di input	35
5.2	Comandi di sezionamento del documento	43
5.3	Loghi particolari	47
5.4	Accenti e caratteri speciali	48
6.1	Tabella composta scorrettamente	60
6.2	Tabella composta secondo le regole	60
6.3	Tabella ottenuta con il pacchetto tabularx	62
6.4	Tabella con allineamento alla virgola	63
6.5	Tabella con font di dimensione ridotta	64
6.6	Opzioni del pacchetto graphicx	67
6.7	Caratteri di trasferimento	71
7.1	Lettere greche minuscole	82
7.2	Spazi in modo matematico.	84

7.3	Gli operatori predefiniti	86
7.4	Lettere greche minuscole	98
7.5	Lettere greche maiuscole	98
7.6	Accenti in modo matematico	98
7.7	Relazioni binarie	99
7.8	Operazioni binarie	99
7.9	Grandi operatori	100
7.10	Delimitatori	100
7.11	Grandi delimitatori	100
7.12	Frecce	101
7.13	Simboli misti	102
7.14	Simboli non matematici	102
7.15	Altri caratteri alfabetici	102
7.16	Altre relazioni binarie	103
7.17	Altre operazioni binarie	103
7.18	Negazioni di simboli	104
7.19	Font matematici	104
9.1	Esempi di voci dell'indice analitico	108
10.1	Comandi per cambiare lo stile dei font	113
10.2	Comandi per cambiare il corpo dei font	114
10.3	I corpi dei font nelle classi standard	115

Capitolo 1

Introduzione

L^AT_EX è un programma di composizione tipografica liberamente disponibile, particolarmente indicato per l'elaborazione di documenti scientifici, ai più elevati livelli di qualità. Lo scopo di questo lavoro, basato sulla mia guida “L'arte di scrivere con L^AT_EX” [21] (cui si rimanda per ogni approfondimento) e rivolto sia ai principianti di L^AT_EX sia a coloro che già lo conoscono, è fornire agli utenti di L^AT_EX di lingua italiana gli elementi essenziali per comporre un documento usando questo potentissimo strumento di scrittura.

Tale obiettivo è perseguito innanzitutto presentando le nozioni fondamentali del programma, nella maniera più chiara e sintetica possibile. Al contempo, vengono forniti svariati esempi e vengono analizzati alcuni problemi tipici incontrati durante la stesura di una pubblicazione scientifica o professionale, specialmente in lingua italiana, indicando le soluzioni che ritengo migliori.

La scelta delle soluzioni adottate deriva sia dalla mia esperienza sia dalle numerose discussioni presenti sul forum del Gruppo Utilizzatori Italiani di T_EX e L^AT_EX (<http://www.guit.sssup.it/>), che resta sempre un eccellente riferimento per tutti i temi trattati in questo documento.

Il testo presume che il lettore possieda una certa familiarità con l'uso del calcolatore e di Internet.

La prassi seguita è quella di non approfondire i vari temi nei dettagli, ma di indirizzare il lettore alla letteratura specifica o ai manuali dei pacchetti suggeriti, quando necessario. Tutte le volte che si cita un pacchetto, non si fornisce una descrizione completa del suo funzionamento, per cui si rimanda alla relativa documentazione, ma si analizzano le opzioni più importanti e se ne suggerisce l'uso.

L'esposizione del lavoro è articolata come segue:

Nel secondo capitolo viene offerta una breve visione d'insieme della storia di L^AT_EX e ne vengono presentate le idee di fondo.

Nel terzo capitolo vengono spiegate le operazioni per installare L^AT_EX sul proprio calcolatore.

Nel quarto capitolo vengono presentate le nozioni fondamentali che permettono di avere una conoscenza di base del funzionamento di L^AT_EX.

Nel quinto capitolo, che estende le conoscenze basilari su $\text{\LaTeX}$ acquisite nel capitolo precedente, vengono fornite le nozioni necessarie per realizzare veri e propri documenti testuali.

Nel sesto capitolo vengono presentati i concetti e gli strumenti fondamentali per comporre e gestire tabelle e figure con $\text{\LaTeX}$.

Nel settimo capitolo viene esplorato uno dei principali punti di forza di $\text{\LaTeX}$, ovvero la composizione di formule matematiche.

Nell'ottavo capitolo viene spiegato come realizzare e gestire una bibliografia con $\text{\LaTeX}$.

Nel nono capitolo vengono illustrate le nozioni essenziali per generare un indice analitico con $\text{\LaTeX}$.

Nel decimo capitolo vengono esposti alcuni suggerimenti su come fare in modo che $\text{\LaTeX}$ produca risultati diversi da quelli predefiniti.

In appendice, infine, vengono descritte sinteticamente le principali norme tipografiche della lingua italiana, utili nella composizione di articoli, tesi o libri.

Se avete idee su argomenti da aggiungere, togliere o modificare in questo documento, o se vi dovesse capitare di notare un errore, sia di battitura sia di sostanza, mi fareste un favore comunicandomelo, così che io possa apportare le opportune correzioni in versioni successive. Mi interessano specialmente i commenti dei principianti di $\text{\LaTeX}$ su quali parti di questo lavoro risultino di facile comprensione e quali invece potrebbero essere spiegate meglio.

È con questo spirito che ho scritto questo lavoro: spero che possiate usare $\text{\LaTeX}$ con il mio stesso piacere.

Capitolo 2

Storia e filosofia

In questo capitolo viene offerta una breve sintesi della storia di $\text{\LaTeX}$ e ne vengono presentate le idee di fondo e le peculiarità.

2.1 Storia

2.1.1 $\text{\TeX}$

$\text{\TeX}$ è un programma di composizione tipografica distribuito con una licenza di software libero, realizzato da Donald E. Knuth, professore di informatica all'università di Stanford.

Knuth iniziò a scrivere il “motore” di tipocomposizione $\text{\TeX}$ nel 1977, allo scopo di esplorare le potenzialità degli strumenti digitali di stampa che a quel tempo stavano iniziando a prendere piede nel campo dell'editoria, con la speranza di poter far regredire la tendenza al deterioramento della qualità tipografica che constatava affliggere i suoi libri e articoli, in particolare il suo capolavoro *The Art of Computer Programming*, in più volumi, ricchi di formule matematiche.

$\text{\TeX}$ è stato pubblicato nel 1982, e negli anni successivi è stato costantemente aggiornato e perfezionato. In particolare, nel 1989 sono stati aggiunti alcuni rilevanti miglioramenti per consentire il supporto dei caratteri a 8 bit e di più lingue. L'ultima revisione di $\text{\TeX}$ è del 2008. $\text{\TeX}$ è rinomato per essere estremamente stabile ed eseguibile su diversi tipi di calcolatori. Il numero di versione di $\text{\TeX}$ converge a π e attualmente è 3.1415926.

The $\text{\TeX}$ book, scritto da Donald Knuth, è il manuale d'uso di $\text{\TeX}$ e uno dei libri più completi su questo programma. Attualmente, $\text{\TeX}$ è un marchio registrato dalla American Mathematical Society ($\mathcal{AMS}$).

2.1.2 Etimologia

Knuth ha nascosto un “trabocchetto” nel nome del suo programma di composizione tipografica: “ $\text{\TeX}$ ” non va letto in alfabeto latino, ma in alfabeto greco (maiuscolo). La lettera “X” di $\text{\TeX}$ non è una “ics” latina, ma un “chi” greco, così come la lettera “E” è un “epsilon”. In lettere minuscole, $\text{\TeX}$ si scriverebbe $\tau\epsilon\chi$.

In un ambiente ASCII, il logo “ $\text{\TeX}$ ” si rende con le maiuscole/minuscole: $\text{\TeX}$.

2.1.3 $\text{\LaTeX}$

$\text{\LaTeX}$ è un programma di composizione tipografica liberamente disponibile, realizzato da Leslie Lamport, che si serve di $\text{\TeX}$ come motore di tipocomposizione. Si tratta di un programma progettato per automatizzare tutte le operazioni più comuni che coinvolgono la realizzazione di un documento, che consente agli autori di impaginare e stampare il proprio lavoro ai più elevati livelli di qualità tipografica, servendosi di impostazioni di pagina professionali predefinite.

Lamport iniziò a scrivere $\text{\LaTeX}$ alla fine degli anni Settanta, quando $\text{\TeX}$ non era ancora stato pubblicato (Lamport era uno dei collaboratori di Knuth nello sviluppo di $\text{\TeX}$). La prima versione pubblica di $\text{\LaTeX}$ risale al 1985, e da allora il programma è stato continuamente aggiornato e migliorato. Per molti anni il numero della versione è rimasto fissato a 2.09 e le successive revisioni sono state identificate con le loro date.

Nel 1994, $\text{\LaTeX}$ è stato aggiornato da un gruppo di programmatori guidato da Frank Mittelbach, al fine di includere alcuni miglioramenti e di riunire tutte le versioni con estensioni e correzioni che si erano raccolte dopo la pubblicazione di $\text{\LaTeX}$ 2.09. Per distinguere la nuova versione da quella precedente, essa è chiamata $\text{\LaTeX}$ 2_ε; questa documentazione tratta $\text{\LaTeX}$ 2_ε.

La versione 3 di $\text{\LaTeX}$ appare come un progetto a lungo termine; i costanti aggiornamenti di $\text{\LaTeX}$ 2_ε ne costituiscono le tappe di avvicinamento. L'ultima versione pubblica di $\text{\LaTeX}$ è del 2005.

2.2 Filosofia

2.2.1 Composizione sincrona e asincrona

La caratteristica che più differenzia $\text{\LaTeX}$ dagli altri elaboratori di testo è il fatto che per comporre un documento con questo programma bisogna agire *in tempi diversi* per introdurre il testo e per comporlo [2, p. 1].

Nei comuni elaboratori di testo (come per esempio Microsoft Word), l'autore agisce direttamente sul testo già composto, così come appare sullo schermo del suo elaboratore, e ogni sua azione si traduce in un'*immediata* variazione del testo composto. Il tipo di composizione che caratterizza questi programmi viene denominato "composizione sincrona". Perché il programma sia davvero sincrono e il ritardo fra azione e visualizzazione sia trascurabile, la forza del programma deve essere concentrata nella rapidità della presentazione. Tale caratteristica non può che andare a discapito della "perfezione" della composizione, perché questa dipende da un'elaborazione molto più accurata sul testo da comporre. È vero che oggi i programmi di videoscrittura sono estremamente rapidi e che ogni anno la qualità della loro composizione migliora vistosamente, tuttavia il compromesso fra velocità e qualità esiste sempre.

La "composizione asincrona" consiste nell'introdurre (con un editor) il testo da comporre in un file, senza badare al suo aspetto grafico, ma concentrandosi sulla struttura logica del testo, e nell'elaborarlo successivamente con $\text{\LaTeX}$, che agisce da impaginatore.

Durante questo processo può accadere, per esempio, di dover modificare una parola di un capoverso. In questo caso l'intero capoverso viene ricomposto: ciò implica l'ottimizzazione compositiva di tutto il capoverso, non solo un piccolo aggiustamento del punto in cui si è eseguita la modifica. Ecco quindi che la composizione avviene in due tempi, l'introduzione del testo e l'ottimizzazione della composizione, avendo però a disposizione l'intero testo da trattare.

Va da sé che la composizione asincrona assicura una migliore qualità di composizione rispetto a quella sincrona, dal momento che non viene tenuta in nessun conto la velocità di visualizzazione, ma la forza compositiva viene concentrata sulla qualità: poiché tratta il testo nel suo complesso (e non man mano che lo si redige), $\text{\LaTeX}$ può avere una visione d'insieme e fare scelte di impaginazione migliori.

2.2.2 Concentrarsi sul contenuto e non sulla forma

L'idea centrale di Lamport era di creare un linguaggio che permettesse ai suoi utenti di concentrarsi sulla struttura logica del testo e non su quella puramente stilistica.

$\text{\LaTeX}$ infatti permette all'utente di *astrarsi* dai dettagli tipografici con cui verrà composto il proprio documento, definendone la struttura logica, piuttosto che dando informazioni sulla disposizione grafica degli elementi che lo compongono (per esempio, specificando che una struttura è un paragrafo o un elenco, piuttosto che affermare che essa dovrebbe essere composta con un'intestazione in neretto e dello spazio prima e dopo). La trasformazione di un testo sorgente scritto con $\text{\LaTeX}$ in un documento finito viene fatta con l'aiuto di strumenti (classi, pacchetti, ...) che, idealmente, permettono cambiamenti anche radicali dell'aspetto, senza modificare il sorgente.

Il file prodotto con l'editor contiene dunque non solo il testo in senso stretto, ma anche le istruzioni di "marcatura" logica (in inglese *mark-up*) che permettono a $\text{\LaTeX}$ di sapere che cosa sta componendo, in modo da eseguirne la composizione secondo le direttive dello stile del documento che sta elaborando. Il testo prodotto mediante l'editor di testo è quindi, in effetti, un codice scritto in un linguaggio di programmazione, che contiene sia il testo da comporre, sia le istruzioni per comporlo.

Un esempio

Per dare un'idea di com'è strutturato un documento elaborato con $\text{\LaTeX}$, di seguito è riportato un frammento di testo sorgente.

```
Due matrici  $n \times n$  complesse  $A$  e  $B$  si dicono simili
se esiste una matrice  $n \times n$  invertibile  $T$  tale che
\begin{equation}
B = T^{-1}AT.
\end{equation}
```

Il testo sorgente con le sue istruzioni viene compilato con $\text{\LaTeX}$, che (attraverso $\text{\TeX}$) produce il documento tipocomposto (in inglese, *typeset*). Se il risultato non è soddisfacente, non si può modificare direttamente il documento composto, ma si corregge il testo sorgente e si ricompila.

Di seguito è riportato un esempio di testo sorgente (sulla sinistra) con accanto il relativo documento composto.

```
Due matrici  $n \times n$ 
complesse  $A$  e  $B$  si dicono
 $\emph{simili}$  se esiste una
matrice  $n \times n$ 
invertibile  $T$  tale che

$$B = T^{-1}AT.$$

```

Due matrici $n \times n$ complesse A e B si dicono *simili* se esiste una matrice $n \times n$ invertibile T tale che

$$B = T^{-1}AT. \quad (2.1)$$

Nei capitoli successivi verranno spiegate tutte le istruzioni usate nell'esempio. Tuttavia anche il lettore con pochi rudimenti di inglese capisce facilmente quello che il linguaggio di marcatura ha specificato.

Capitolo 3

Installare L^AT_EX

In questo capitolo vengono spiegate le operazioni per installare L^AT_EX sul proprio calcolatore. Le distribuzioni di L^AT_EX prese in considerazione sono MiK_TE_X (per Windows) e T_EX Live (multiplatforma), che sono ottime, diffuse e gratuite. Si distinguono tre casi:

- sistema operativo Windows (da Windows 2000 in poi);
- sistema operativo Mac OS X (dalla versione 10.3 in poi);
- sistema operativo Linux.

3.1 Introduzione

Come anticipato nel capitolo precedente, gli elementi essenziali per scrivere un documento con L^AT_EX sono:

- un editor di testo con cui creare il file sorgente `.tex` contenente il testo da comporre;
- L^AT_EX, che elabora il file `.tex` creato con l'editor e produce il documento tipocomposto (in formato `.pdf` o `.dvi`);
- un programma per visualizzare il documento (`.pdf` o `.dvi`) elaborato da L^AT_EX.

È bene sottolineare che nonostante, in teoria, sia possibile usare qualsiasi editor di testo, per gestire al meglio i file sorgente è opportuno servirsi di un editor specifico per L^AT_EX. In questo modo, fra l'altro, basta premere un pulsante dell'editor per compilare con L^AT_EX e, se non ci sono errori, per attivare anche il programma per visualizzare il documento; a ogni successiva compilazione, la finestra del programma impiegato per visualizzare il documento viene di regola aggiornata automaticamente, per cui si ha una compilazione “quasi sincrona”. Esistono numerosi editor specifici per L^AT_EX, fra cui è possibile scegliere quello più adatto alle proprie esigenze.

Una *distribuzione* di L^AT_EX è una raccolta di file e programmi che contiene quanto serve per comporre un documento con L^AT_EX. Generalmente una distribuzione

contiene, oltre al sistema $\text{\LaTeX}$ (e ai programmi principali a esso connessi) uno o più editor e uno o più programmi per visualizzare i documenti elaborati da $\text{\LaTeX}$. Le distribuzioni di $\text{\LaTeX}$ possono essere installate da dvd oppure da Internet.

Va notato che il Gruppo Utilizzatori di $\text{\TeX}$ e $\text{\LaTeX}$ (<http://www.guit.sssup.it/guit/>) invia a tutti i suoi soci il dvd “ $\text{\TeX}$ Collection”, che contiene le distribuzioni di $\text{\LaTeX}$ per i principali sistemi operativi; le distribuzioni presentate in questo capitolo possono essere ritrovate in quel disco; ciò è molto utile se non si dispone di una connessione a Internet veloce.

Per installare una distribuzione di $\text{\LaTeX}$ da Internet, il luogo di riferimento è CTAN (*Comprehensive $\text{\TeX}$ Archive Network*, “rete di archivi completi per $\text{\TeX}$ ”, <http://www.ctan.org/>); qui è possibile rintracciare tantissimo software e materiale che riguarda $\text{\LaTeX}$, per esempio pacchetti, programmi, stili, font.¹

3.2 Installazione per Windows

Su Windows sono disponibili le distribuzioni MiK $\text{\TeX}$ e $\text{\TeX}$ Live: sono entrambe eccellenti, ma mentre MiK $\text{\TeX}$ è specifica per Windows e facile da installare e gestire, $\text{\TeX}$ Live è multiplatforma e più adatta agli utenti esperti.

3.2.1 La distribuzione MiK $\text{\TeX}$

La distribuzione MiK $\text{\TeX}$ è specifica per il sistema operativo Windows. La sua installazione, simile a quella di ogni altro programma per Windows, non presenta particolari difficoltà (bastano pochi clic del mouse). È possibile scaricare MiK $\text{\TeX}$ dal sito <http://miktex.org/>. Dalla pagina di download, si scarica il file `setup.exe`: questo piccolo programma va eseguito una prima volta per scaricare la distribuzione e una seconda volta per effettuare l’installazione vera e propria. È sufficiente seguire le semplici istruzioni a video. Sono disponibili due versioni di MiK $\text{\TeX}$, “di base” (*basic*) e “completa” (*complete*), che differiscono unicamente per la disponibilità di pacchetti, che con MiK $\text{\TeX}$ si possono gestire in modo particolarmente semplice (vedi il paragrafo 4.5.4 a pagina 25). La versione completa è ovviamente la scelta consigliata. È bene tener presente che la distribuzione MiK $\text{\TeX}$ non è automaticamente predisposta per lavorare con tutte le lingue che $\text{\LaTeX}$ è capace di gestire (italiano compreso); per farlo è necessario eseguire la semplice procedura descritta nel paragrafo 5.1.1 a pagina 33.²

¹La *rete* CTAN è un insieme di siti dislocati in tutto il mondo, tutti uguali fra di loro. Il vantaggio di collegarsi a un server CTAN più vicino, che contiene una copia integrale (*mirror*, “specchio”) di un sito CTAN “originale” (che può essere sovraffollato), è di poter disporre, solitamente, di una connessione più veloce. In Italia esiste, per esempio, il server dell’Università Tor Vergata di Roma (<ftp.uniroma2.it>), cui si viene automaticamente indirizzati quando si scaricano da CTAN le distribuzioni MiK $\text{\TeX}$ o $\text{\TeX}$ Live. Gli indirizzi dei server CTAN disponibili in Rete sono indicati nella pagina Web <http://www.ctan.org/tex-archive/CTAN.sites>.

²È disponibile anche la distribuzione pro $\text{\TeX}$ t (presente su <http://www.tug.org/protext/> o sul dvd “ $\text{\TeX}$ Collection”), che è basata sulla distribuzione MiK $\text{\TeX}$, alla quale sono stati affiancati gli editor WinEdt e $\text{\TeX}$ nicCenter e i programmi accessori GSview e Ghostscript. La procedura di installazione di pro $\text{\TeX}$ t è molto semplice e ben documentata.

3.2.2 La distribuzione T_EX Live per Windows

Dal 2008, la distribuzione T_EX Live, nata in ambiente Unix/Linux, è disponibile anche per Windows. Nonostante si tratti di una distribuzione robusta e affidabile (e, a differenza di MiK_TE_X, automaticamente configurata per lavorare con tutte le lingue che L^AT_EX è capace di gestire, italiano compreso), le procedure di installazione e di manutenzione della T_EX Live per Windows non sono semplici come quelle di MiK_TE_X, che è pensata appositamente per gli utenti di Windows: bisogna studiare la documentazione (in inglese, <http://tug.org/tex/texlive/windows.html>) e può essere necessario servirsi della linea di comando. Allo stato attuale, se ne consiglia l'uso solo agli utenti esperti.

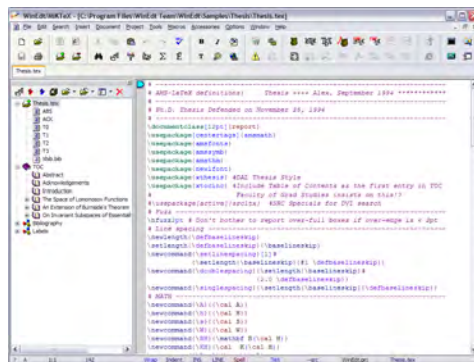
3.2.3 Editor per Windows

Su Windows sono disponibili numerosi editor specifici per L^AT_EX. Fra i più diffusi vi sono T_EXnicCenter, WinEdit e L^ED, che sono tutti facili da installare e usare. Quale impiegare è, in definitiva, questione di gusti: si consiglia all'utente di provarli e di scegliere quello più adatto alle proprie esigenze.

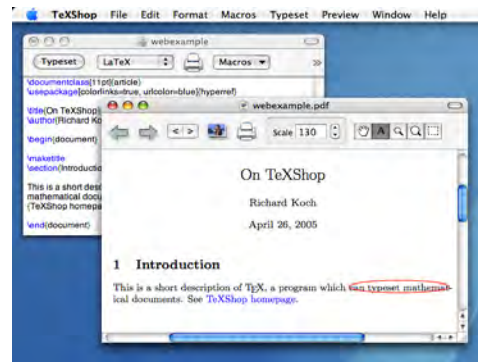
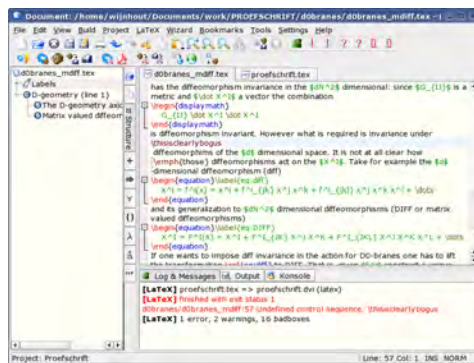
T_EXnicCenter

T_EXnicCenter è un editor per Windows distribuito gratuitamente sotto licenza GNU. È abbastanza simile a WinEdt, rispetto al quale presenta un'interfaccia e delle funzioni più semplici. Prima di iniziare l'installazione di T_EXnicCenter, si consiglia di aver installato una delle distribuzioni di L^AT_EX disponibili per Windows; in caso contrario la configurazione del programma dovrà essere posticipata al momento in cui una distribuzione si renderà disponibile all'interno del sistema. Il programma si scarica dal sito <http://www.texniccenter.org/>. Al primo avvio dell'editor partirà la procedura di configurazione automatica, durante la quale è necessario specificare se si sta usando MiK_TE_X oppure un'altra distribuzione.

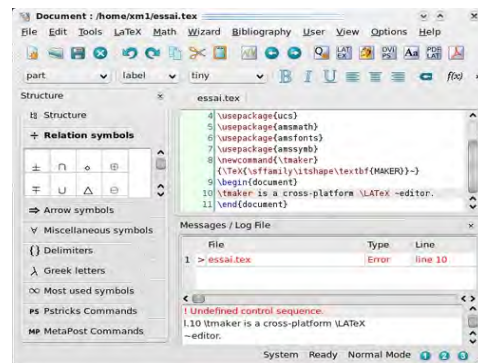
Il controllo ortografico in lingua italiana: T_EXnicCenter usa lo stesso dizionario di Open Office, pertanto è possibile scaricare il dizionario dalla pagina dedicata ai download del sito di Open Office (<http://it.openoffice.org/>) o, in alternativa, dal sito di G_UI_T (http://www.guit.sssup.it/installazione/win_editor.php). Dopo aver decompresso l'archivio e aver copiato i due file `it_IT.aff` e `it_IT.dic` nella cartella `... \TeXnicCenter \Language \` (la posizione esatta della cartella che contiene il programma dipende dalla scelta fatta dall'utente durante l'installazione). A questo punto si avvia T_EXnicCenter e, tramite il menu `Tools → Options → Spelling`, si abilitano le funzioni `language:it`, `dialect:IT`, poi si inserisce in `personal dictionary` la voce `... \TeXnicCenter \Language \it_IT.dic` (anche in questo caso il percorso che individua la cartella contenente il programma dipende dalla scelta specificata durante l'installazione).



(a) WinEdt (per Windows).

(b) $\text{\TeX}$ Shop (per Mac).

(c) Kile (per Linux).



(d) Texmaker (multiplatforma).

Figura 3.1: Alcuni editor per $\text{\LaTeX}$.

WinEdt

WinEdt è un ottimo editor shareware per Windows (che permette di gestire anche altri tipi di file, per esempio .html). La sua installazione non presenta alcuna difficoltà. Il programma si scarica dal sito <http://www.winedt.com/>.

Il controllo ortografico in lingua italiana: Per usufruire della funzione di controllo ortografico è necessario installare nel programma il dizionario per la lingua italiana. A tal fine, è sufficiente creare la cartella `it\` in `C:\Programmi\WinEdt Team\WinEdt\Dict\`, scaricare dal sito di $\text{\LaTeX}$ il dizionario (http://www.guit.sssup.it/installazione/win_editor.php) e copiarlo nella cartella appena creata. Dal pannello Dictionaries del menu Options → Dictionary Manager di WinEdt, si seleziona la voce Your custom dictionaries e si inserisce nel campo Definitions la voce `%B\Dict\it\It.dic`. Per abilitare l'uso del dizionario è sufficiente spuntare nella casella relativa la voce Enabled, verificando che nel campo Options siano attivati i comandi Load On Start, Save On Exit, Add New Words e Use for Completions.

L^AEd

L^AEd (L^AT_EX Editor) è un editor gratuito per Windows, specifico per L^AT_EX (ma che permette di gestire anche altri tipi di file, per esempio .html, Pascal o Perl). Facile da installare e usare, è dotato di un'interfaccia grafica (disponibile anche in lingua italiana) comodamente configurabile e particolarmente ricca di funzioni. Si scarica dal sito <http://www.latexeditor.org/>. Per usufruire della funzione di controllo ortografico è necessario scaricare, dal menu Configuration → Add-ons, il dizionario della lingua italiana; successivamente, dal menu Configuration → Options → Spellchecking → Dictionaries, si abilita la funzione selezionando la voce Italian.

3.3 Installazione per Mac

Installare L^AT_EX su Mac è semplicissimo: la distribuzione T_EX Live, che su Mac è nota anche come MacT_EX, si installa come qualsiasi altro programma per Mac.

3.3.1 La distribuzione MacT_EX

Per installare la distribuzione T_EX Live su Mac è sufficiente scaricare il file Mac-TeX.dmg da <http://www.tug.org/mactex/>. All'interno del file è contenuta l'intera distribuzione, assieme a svariati programmi utili, come per esempio T_EXShop. Una volta scaricato il file, basta aprirlo ed eseguire le semplicissime istruzioni a video.

Fin dalla prima installazione, la distribuzione è automaticamente predisposta per lavorare con tutte le lingue che L^AT_EX è capace di gestire (italiano compreso). Il sistema operativo è già dotato dei programmi accessori per visualizzare il risultato della composizione e per procedere alla stampa. Anche la gestione dei pacchetti è particolarmente semplice (vedi il paragrafo 4.5.4 a pagina 25).

3.3.2 Editor per Mac

Su Mac sono disponibili diversi editor specifici per L^AT_EX: T_EXShop è un classico, semplicissimo da installare e usare.

T_EXShop

T_EXShop è un ottimo editor gratuito per Mac, coperto da licenza GNU. Nella compilazione, T_EXShop impiega PDF_LA_TE_X (non producendo quindi file D_VI) e usa un apposito programma per visualizzare e stampare i documenti compilati, in formato PDF. L'editor è già dotato del dizionario italiano e inglese. Il sito Web del progetto è <http://www.uoregon.edu/~koch/texshop/>.

3.4 Installazione per Linux

L'installazione di un sistema L^AT_EX su Linux non presenta, di regola, difficoltà eccessive. Spesso L^AT_EX risulta addirittura già installato assieme al sistema operativo. In ogni caso, la distribuzione T_EX Live può venire installata come qualunque altro programma di Linux.

3.4.1 La distribuzione $\text{\TeX}$ Live per Linux

Su Linux, la distribuzione $\text{\TeX}$ Live (<http://www.tug.org/texlive/>) può essere installata dalla linea di comando (scelta comune fra gli utenti di questa piattaforma) oppure avvalendosi di interfacce grafiche, come per esempio $\text{\KDE}$. Generalmente non c'è altro da fare, salvo, se si installa una versione di $\text{\TeX}$ Live precedente alla 2008, eseguire il programma `texconfig` per configurare il sistema per l'uso della lingua italiana (vedi il paragrafo 5.1.1 a pagina 33). Neppure la gestione dei pacchetti presenta troppe difficoltà (vedi il paragrafo 4.5.4 a pagina 25).

Se si usa una distribuzione di Linux basata su Debian (fra cui Ubuntu e le sue varianti), l'installazione di $\text{\TeX}$ Live 2008 presenta qualche problema; nell'attesa che la prossima $\text{\TeX}$ Live 2009 renda più semplice l'installazione, per installare $\text{\TeX}$ Live 2008 nei sistemi Debian è possibile seguire la procedura descritta in <http://texblog.net/latex-archive/linux/kile-texlive-2008-equivs/> (in inglese); in alternativa, si può optare per la $\text{\TeX}$ Live 2007, la cui installazione non presenta difficoltà.

3.4.2 Editor per Linux

La distribuzione $\text{\TeX}$ Live per Linux contiene anche diversi editor fra cui scegliere quello più adatto alle proprie esigenze. Si può usare l'onnipresente Emacs con l'estensione `auctex`; in alternativa, se lo si trova troppo complicato, si può installare Kile, facile da usare.

Kile

Kile è un ottimo editor con licenza GNU per $\text{\KDE}$. Incluso nella distribuzione $\text{\TeX}$ Live, si avvale di un'interfaccia grafica che include le funzioni a cui si ricorre più frequentemente durante la scrittura di un documento in $\text{\LaTeX}$. Particolare attenzione è stata posta alla possibilità di interagire con altri programmi (come Xfig e Gnuplot). Le versioni di Kile allegate alla distribuzione $\text{\TeX}$ Live sono già dotate del dizionario italiano e inglese. Il sito Internet del progetto è <http://kile.sourceforge.net/>.

3.5 Altri programmi utili

Lavorando con $\text{\LaTeX}$, è possibile usare un programma specializzato per ciascuna operazione di composizione, in modo da servirsi sempre del prodotto migliore per ciascuna fase di elaborazione del documento. I seguenti sono alcuni utili programmi accessori.

Adobe Reader è un programma multiplatforma e gratuito prodotto da Adobe, che consente di visualizzare e stampare i file nel formato PDF. Si scarica da <http://www.adobe.com/it/>.

Ghostview è un programma gratuito per Unix/Linux, che permette di visualizzare file PS, EPS e PDF. Permette, fra l'altro, di convertire file dal formato EPS e PS al formato PDF. Esiste anche un programma simile, GSview, disponibile

per Windows e Unix/Linux. Entrambi i programmi si scaricano dal sito <http://pages.cs.wisc.edu/~ghost/> e richiedono Ghostscript (disponibile allo stesso indirizzo Internet).

Gimp è un programma gratuito nato per la piattaforma Linux e ora disponibile anche per le piattaforme Windows e Mac. Consente di modificare le immagini bitmap di svariati formati e di registrarle in uno qualunque dei numerosi formati che è in grado di gestire. Si scarica da <http://www.gimp.org/>.

Questi programmi accessori sono utili di per sé e quindi possono essere impiegati anche indipendentemente da $\text{\LaTeX}$, tuttavia risultano molto utili per svolgere egregiamente certe funzioni che solo un programma specializzato sa fare, certamente meglio di quanto potrebbe fare un programma generico dalle “troppe” funzioni.

3.6 $\text{\LaTeX}$ e $\text{\PDF\LaTeX}$

È bene che il lettore sappia che con “ $\text{\LaTeX}$ ” si intende non solo il programma $\text{\LaTeX}$ vero e proprio, che produce in output file $\text{\DVI}$ (*DeVice Independent*, “indipendenti dal dispositivo”), ma anche il programma $\text{\PDF\LaTeX}$, il cui formato di uscita è direttamente il $\text{\PDF}$ (*Portable document format*, il formato di file più versatile per la stampa e la distribuzione elettronica dei documenti), senza che siano necessarie conversioni di alcun genere.

Entrambi i formati hanno pro e contro e vengono usati con scopi differenti [16, p. 34]. Il formato $\text{\DVI}$ permette la *ricerca diretta* (facendo doppio clic sul codice all’interno dell’editor, il programma di visualizzazione del $\text{\DVI}$ trova il rispettivo output) e *inversa* (facendo doppio clic all’interno del programma di visualizzazione del $\text{\DVI}$, il cursore viene posizionato sul rispettivo codice all’interno dell’editor), che possono essere utili in fase di elaborazione del documento. Tuttavia la maggioranza dei visualizzatori di $\text{\DVI}$ non visualizza correttamente gli effetti di alcuni comandi che modificano le figure e inoltre non sfrutta il pacchetto microtype. Il $\text{\PDF}$, invece, non offre in generale la ricerca diretta e inversa — in realtà, su macchine Mac che usano la distribuzione $\text{\MacTeX}$ e su PC che impiegano SumatraPDF (<http://william.famille-blum.org/software/sumatra/>) per visualizzare i $\text{\PDF}$, la ricerca è eseguibile in entrambi i sensi anche su file $\text{\PDF}$. Tuttavia, il $\text{\PDF}$ visualizza correttamente tutti gli effetti del pacchetto graphicx, sfrutta il pacchetto microtype, è un formato molto diffuso anche tra coloro che non usano $\text{\LaTeX}$, permette di attivare i riferimenti ipertestuali forniti dal pacchetto hyperref e consente di restringere l’accesso al documento per mezzo di password: per esempio, è possibile limitare l’accesso al file, limitarne la stampa (impedirla, oppure consentirla solo a bassa risoluzione) e limitarne le modifiche (estrazione di testo, estrazione o eliminazione di pagine, ...).

Si noti che la creazione di un $\text{\PDF}$ può essere effettuata nei due seguenti modi:

- la conversione di un file $\text{\DVI}$ in $\text{\PDF}$ (di regola tramite l’applicazione in sequenza dei programmi $\text{\dvips}$ e $\text{\ps2pdf}$ o mediante il programma $\text{\dvi2pdfmx}$, operazione per la quale è generalmente sufficiente cliccare su un apposito bottone dell’editor usato);

- la compilazione diretta del sorgente `.tex` con $\text{\pdfLaTeX}$.

Tuttavia, per sfruttare appieno le potenzialità del formato $\text{\pdf}$ è necessario usare $\text{\pdfLaTeX}$: per esempio, il formato $\text{\pdf}$ permette di ottenere i riferimenti ipertestuali (*hyperlink*), i segnalibri (*bookmark*) e le immagini in miniatura (*thumbnail*) che non vengono prodotti nella conversione a partire da file $\text{\dvi}$. Il programma $\text{\pdfLaTeX}$ è presente in tutte le più importanti distribuzioni di $\text{\LaTeX}$ ed è gestito dai principali editor di testo dedicati a $\text{\LaTeX}$.

In generale, i file sorgente elaborabili con $\text{\LaTeX}$ o con $\text{\pdfLaTeX}$ sono identici, o possono essere resi tali. Pertanto quando si parla di un file sorgente elaborabile con $\text{\LaTeX}$, si intende che il file è elaborabile indifferentemente anche da $\text{\pdfLaTeX}$.

Tuttavia, esistono casi in cui un file conforme alla grammatica di $\text{\LaTeX}$ può risultare incompatibile con $\text{\pdfLaTeX}$. Ciò succede raramente, e di solito dipende dall'uso di pacchetti particolarmente specializzati. Se possibile, sarebbe meglio evitare tali pacchetti, ricorrendo eventualmente a pacchetti alternativi, ugualmente adeguati per la particolare funzione per la quale essi vengono invocati [2, p. 21]. Questa soluzione funziona nella stragrande maggioranza dei casi.

Tipicamente, l'incompatibilità fra $\text{\LaTeX}$ e $\text{\pdfLaTeX}$ non dipende dal file sorgente, ma dai file inclusi; ciò accade di regola con i file che contengono delle immagini ($\text{\LaTeX}$ richiede immagini esclusivamente in formato $\text{\eps}$, mentre $\text{\pdfLaTeX}$ accetta immagini in formato $\text{\pdf}$, se vettoriali, oppure $\text{\jpg}$ o $\text{\png}$, se bitmap). Nel paragrafo 6.2 a pagina 64 sono spiegate le soluzioni da adottare per risolvere questo problema.

In definitiva, si consiglia di usare il formato $\text{\dvi}$ durante la stesura di un documento, solo se si usa Windows o Linux e si ha la necessità di usare le funzioni di ricerca diretta e inversa. Per tutti gli altri casi, tra cui la stampa e la distribuzione del documento in formato elettronico, si consiglia di usare $\text{\pdfLaTeX}$.

Capitolo 4

Le basi

In questo capitolo vengono presentate le nozioni fondamentali che permettono di avere una conoscenza di base del funzionamento di $\text{\LaTeX}$; procedendo nella lettura, sarà possibile estendere e arricchire queste conoscenze con nuove informazioni.

4.1 Il nostro primo documento con $\text{\LaTeX}$

In questo paragrafo realizzeremo con $\text{\LaTeX}$ un semplice documento, imparando come:

- scrivere il sorgente del documento;
- compilarlo;
- trovare e correggere gli errori che può contenere;
- visualizzare il documento;
- stamparlo.

4.1.1 La scrittura del sorgente

Innanzitutto, creiamo una cartella prova nella quale inserire tutti i file del documento. Ciò fatto, con l'editor di testo prescelto, che di regola sarà un editor specifico per $\text{\LaTeX}$, ma che in teoria può essere un editor qualunque, scriviamo il seguente testo (le istruzioni in esso contenute verranno spiegate nei paragrafi successivi):

```
\documentclass[a4paper]{article}
\usepackage[latin1]{inputenc}
\usepackage[italian]{babel}
\usepackage[T1]{fontenc}
\begin{document}
Ecco il mio primo documento con \LaTeX.
\end{document}
```

Registriamo il file con il nome `primo.tex` (per convenzione, i file sorgente di $\text{\LaTeX}$ hanno l'estensione `.tex`).

4.1.2 La compilazione

Se ci siamo serviti di un editor specifico per $\text{\LaTeX}$, per compilare il sorgente basta premere l'apposito pulsante dell'editor. $\text{\LaTeX}$ mostra man mano quello che sta facendo, emettendo un messaggio di questo tipo:

```
This is pdfTeX, Version 3.141592 (Web2C 7.5.6)
(primo.tex
LaTeX2e <2005/12/01>
Babel <v3.8> and hyphenation patterns for english, italian, loaded.
(/usr/local/texlive/2007/texmf-dist/tex/latex/base/article.cls
Document Class: article v1.4f Standard LaTeX document class
(/usr/local/texlive/2007/texmf-dist/tex/latex/base/inputenc.sty
(/usr/local/texlive/2007/texmf-dist/tex/latex/base/latin1.def))
(/usr/local/texlive/2007/texmf-dist/tex/generic/babel/babel.sty
(/usr/local/texlive/2007/texmf-dist/tex/generic/babel/italian.ldf
(/usr/local/texlive/2007/texmf-dist/tex/generic/babel/babel.def)))
(/usr/local/texlive/2007/texmf-dist/tex/latex/base/fontenc.sty
(/usr/local/texlive/2007/texmf-dist/tex/latex/base/t1enc.def))
[1] (primo.aux) )
Output written on primo.pdf (1 page, 17114 bytes).
Transcript written on primo.log.
```

Va tenuto presente che la pressione del pulsante dell'editor è equivalente a dare dalla linea di comando (dopo essersi portati nella stessa cartella del file) l'istruzione `latex primo.tex`, se si compila con $\text{\LaTeX}$, o `pdflatex primo.tex`, se si compila con $\text{\pdfLaTeX}$.¹

In questo modo, si è consapevoli del fatto che è $\text{\LaTeX}/\text{\pdfLaTeX}$ (e *non* l'editor) il programma che elabora il sorgente `.tex`, che emette i messaggi e che produce il `.dvi` o il `.pdf`, lavorando “dietro le quinte”. La compilazione non è dunque semplicemente «il tempo che bisogna attendere per vedere il `.dvi` o il `.pdf`», ma il processo durante il quale il programma comprende la nostra volontà e la trasforma per noi in un file (tipo)grafico. È questo processo che viene avviato dalla pressione del tasto opportuno nell'editor usato.

4.1.3 La correzione degli errori

Se il codice sorgente contiene un errore, appare un messaggio di questo tipo:

```
! Undefined control sequence.
1.6 Questo è il mio primo documento con \latex
?
```

Quando trova un errore, $\text{\LaTeX}$ indica:

- la natura dell'errore (in questo caso `Undefined control sequence`, “sequenza di controllo non definita”, ovvero “comando sconosciuto”);

¹Come avviare l'interfaccia a linea di comando (che su Windows si chiama “prompt dei comandi”, su Mac “Terminale” e su Linux “Terminale” o “Console”) dipende dal sistema operativo usato. Su Windows essa si raggiunge da *Avvio/Programmi/Accessori*, su Mac ci si arriva dalla cartella *Applicazioni/Utility*, mentre su Linux come avviarla dipende dalla distribuzione usata.

- la riga in cui si trova l'errore (1.6) ;
- il comando sconosciuto (il comando giusto è `\LaTeX`, non `\latex`; bisogna fare attenzione alle maiuscole!).

Per interrompere la compilazione è sufficiente dare `x` seguito da `Invio`. Dopo aver corretto il sorgente, si registra il file e si ricompila.

4.1.4 La visualizzazione

Se si usa un editor specifico per L^AT_EX e se non ci sono errori, il programma per visualizzare il documento (in formato `.dvi` se si è compilato con L^AT_EX, `.pdf` se si è compilato con PDFL^AT_EX) viene attivato automaticamente.²

Si aprirà una finestra che mostra il documento così come verrà stampato. Con i pulsanti situati nella finestra del programma è possibile, fra l'altro:

- spostarsi all'interno del documento;
- modificarne l'ingrandimento.

4.1.5 La stampa

Per stampare il documento, per la ragioni esaminate nel paragrafo 3.6 a pagina 13 è opportuno compilare con PDFL^AT_EX; ciò fatto, per eseguire la stampa basta premere l'apposito pulsante del programma usato per visualizzare il `.pdf`.

4.2 I file sorgenti di L^AT_EX

Un file sorgente di L^AT_EX è un file di testo in ASCII puro. Può essere creato con qualunque editor di testi e contiene sia il testo del documento sia i comandi che comunicano a L^AT_EX come impostare il formato del testo.

4.2.1 Spazi

Nel testo sorgente lo spazio e la tabulazione sono trattati indifferentemente come "spazio" da L^AT_EX. *Più spazi consecutivi* sono considerati come *uno* spazio. Lo spazio all'inizio di una riga viene ignorato, e una sola interruzione di riga è trattata come uno spazio.

Una riga vuota tra due righe di testo indica la fine di un capoverso. *Più righe vuote* di seguito sono considerate alla stessa maniera di *una* riga vuota. Il testo che segue è un esempio: nella parte sinistra vi è il codice sorgente, e a destra è visualizzato il risultato in stampa.

²In alternativa, se si opera dalla linea di comando, per visualizzare il documento è sufficiente, dopo essersi portati nella stessa cartella del file, dare l'istruzione `<nome del programma> primo.dvi` oppure `<nome del programma> primo.pdf`, dove `<nome del programma>` è il nome del lettore di `.dvi` o di `.pdf` usato.

Non ha alcuna importanza se si mettono uno o tanti spazi dopo una parola.

Una riga vuota fa iniziare un nuovo capoverso.

Non ha alcuna importanza se si mettono uno o tanti spazi dopo una parola.

Una riga vuota fa iniziare un nuovo capoverso.

4.2.2 I caratteri speciali

Ogni programma di composizione tipografica riceve dall'utente:

- il testo;
- i comandi per impostare il formato del testo.

I programmi di composizione sincrona prendono i comandi da appositi menu: l'utente vede l'effetto, mentre la codifica interna rimane nascosta. Con $\text{\LaTeX}$, invece, si inseriscono i comandi insieme con il testo.

Per distinguere i comandi dal testo, $\text{\LaTeX}$ interpreta in modo particolare certi caratteri. I caratteri speciali di $\text{\LaTeX}$ sono i seguenti:

$\backslash \{ \} \% \$ _ \& \# \wedge \sim$

$\backslash$

La barra rovescia (o *backslash*) inizia i *comandi* di $\text{\LaTeX}$.

$\{ \}$

Le parentesi graffe racchiudono i *gruppi*.

$\%$

Il simbolo di percento inizia i commenti.

$\$$

Il dollaro delimita le formule matematiche *in corpo*.

$_$

La sottolineatura (o *underscore*) indica i pedici nelle formule.

$\&$

La “e” commerciale (o *ampersand*) serve da separatore nelle tabulazioni.

$\#$

Il cancelletto (o *diesis*, o *hash*) indica l'argomento quando si definiscono nuovi comandi.

$\wedge$

Il cappuccio (o *caret*) indica l'esponente nelle formule matematiche.

$\sim$

La tilde produce uno spazio insecabile.

Tabella 4.1: Codici ASCII di alcuni caratteri di uso frequente.

Simbolo	Codice
`	96
{	123
}	125
~	126

Ogni utente di L^AT_EX deve imparare a localizzare sulla tastiera i caratteri speciali. Purtroppo la posizione dei caratteri speciali non è standardizzata: cambia, fra l'altro, con il sistema operativo. A volte può essere necessario premere più tasti insieme, o anche digitare codici numerici.

Chi scrive con una tastiera italiana può avere problemi a digitare i quattro caratteri { ` ~ }, che non compaiono su alcun tasto; bisogna fare attenzione a distinguere il carattere ` (virgoletta aperta, accento grave) da ' (usato per inserire l'apostrofo, la virgoletta chiusa e l'accento acuto).

- Se si usa una tastiera italiana PC e si adopera Windows, i caratteri { ` ~ } possono essere ottenuti tenendo premuto il tasto ALT e digitando il codice ASCII relativo con il tastierino numerico (vedi la tabella 4.1). Le parentesi graffe aperte e chiuse si ottengono anche premendo rispettivamente ALT GR + MAIUSCOLE + “è” e ALT GR + MAIUSCOLE + “+” (ovvero come se si dovessero fare delle parentesi quadre “maiuscole”).
- Se si usa una tastiera italiana Mac, la tilde si ottiene con OPZIONE + 5, la virgoletta aperta con OPZIONE + 9 e le parentesi graffe aperte e chiuse con OPZIONE + MAIUSCOLE + “è” e OPZIONE + MAIUSCOLE + “+”, rispettivamente.
- Su Linux, la tilde si ottiene con ALT GR + ì, la virgoletta aperta con ALT GR + ' e le parentesi graffe aperte e chiuse con ALT GR + 7 (oppure ALT GR + MAIUSCOLE + “è”) e ALT GR + 8 (oppure ALT GR + MAIUSCOLE + “+”), rispettivamente.

I caratteri speciali (a eccezione della barra rovescia, del cappuccio e della tilde) si possono ottenere in stampa facendoli precedere nel testo sorgente da \.

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

Il carattere di barra rovescia \ non può essere immesso aggiungendo un altro \ davanti a esso (la sequenza \\ si usa infatti per le interruzioni di riga nei casi che lo prevedono): a tal fine si impiega il comando \textbackslash.

Il cappuccio e la tilde si ottengono facendoli precedere da \ e seguire da due parentesi graffe {}, perché \^ e \~ sono comandi per gli accenti (vedi il paragrafo 5.7.3 a pagina 47) e richiedono un argomento. In alternativa, è possibile usare i comandi \textasciicircum e \textasciitilde, che producono simboli leggermente diversi.

```
\^{ } \textasciicircum \^      ^ ^
\~{ } \textasciitilde ~ ~      ~ ~
```

4.2.3 I comandi

In $\text{\LaTeX}$, i comandi sono composti da una barra rovescia $\backslash$ più altri caratteri. Nei comandi, maiuscole e minuscole sono distinte. Ci sono due tipi di comandi:

- comandi che iniziano con una barra rovescia $\backslash$ seguita da un nome fatto di sole lettere alfabetiche; il comando finisce con il primo carattere non alfabetico (esempi: $\backslash\text{TeX}$, $\backslash\text{end}\{\text{document}\}$, $\backslash\text{documentclass}$).
- comandi che iniziano con una barra rovescia $\backslash$ seguita da esattamente un carattere non alfabetico (esempi: $\backslash\{$, $\backslash\}$, $\backslash\%$, $\backslash\$$, $\backslash_$, $\backslash\&$, $\backslash\#$, $\backslash\sim$).

Gli spazi dopo i comandi sono *ignorati* da $\text{\LaTeX}$. Se si desidera ottenere uno spazio dopo un comando, si deve scrivere $\{ \}$ e uno spazio, oppure scrivere un comando esplicito di spaziatura, come $\backslash_$.

$\backslash\text{Ars}\{ \}$ è la rivista del Gruppo
Utilizzatori Italiani
di $\backslash\text{TeX}\{ \}$ e $\backslash\text{LaTeX}$.

$\text{\textbackslash rsTeX} \text{nica}$ è la rivista del Gruppo
Utilizzatori Italiani di $\text{\TeX}$ e $\text{\LaTeX}$.

Per riprodurre il logo di $\text{\textbackslash rsTeX} \text{nica}$ serve il comando $\backslash\text{Ars}$ del pacchetto `guit`.

Alcuni comandi necessitano di un *argomento* (o *parametro*) che deve essere fornito tra parentesi graffe $\{ \}$ dopo il nome del comando. Certi comandi accettano *parametri opzionali* (facoltativi) che si aggiungono dopo il nome del comando tra parentesi quadre $[]$. I prossimi esempi si servono di alcuni comandi di $\text{\LaTeX}$.

Data odierna: $\backslash\text{today}$.

Data odierna: 1 ottobre 2009.

Sono qui in $\backslash\text{emph}\{\text{dieci}\}$
minuti.

Sono qui in *dieci* minuti.

Il comando $\backslash\text{today}$ indica la data odierna nella lingua corrente. Il comando $\backslash\text{emph}$ serve per evidenziare il testo (vedi il paragrafo 5.11 a pagina 51).

4.2.4 I commenti

Quando, durante l'elaborazione di un file sorgente, $\text{\LaTeX}$ incontra un carattere $\%$ (eccetto $\backslash\%$), ignora il resto della riga, l'interruzione di riga, e tutti gli spazi bianchi all'inizio della riga successiva.

Il carattere $\%$ può essere dunque impiegato per iniziare un pro memoria che non viene stampato. Talvolta il percento si usa per spezzare parole troppo lunghe, o per dividere righe in cui non sono permessi spazi bianchi o interruzioni di riga.

Ecco un $\%$ semplice,
 $\%$ ma istruttivo <----
esempio: Supercal%
 ifragilist%
 ichespiralidoso.

Ecco un esempio: Supercalifragilistiche-
spiralidoso.

Per commenti più lunghi si usa l'ambiente `comment` fornito dall'omonimo pacchetto.

Ecco un altro \begin{comment} semplice, ma utile \end{comment} esempio per includere commenti nel proprio documento.	Ecco un altro esempio per includere commenti nel proprio documento.
--	--

4.3 La struttura del file sorgente

Quando $\text{\LaTeX}$ elabora un file sorgente, si aspetta di seguire una certa struttura. Un testo sorgente di $\text{\LaTeX}$ deve contenere almeno una dichiarazione di classe,

```
\documentclass{\dots}
```

e le dichiarazioni di inizio e fine del documento:

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

Tutto ciò che è compreso tra `\documentclass{\dots}` e `\begin{document}` si dice *preambolo del documento* (o semplicemente *preambolo*) e comprende:

- il caricamento di pacchetti che permettono di estendere le capacità di $\text{\LaTeX}$:
`\usepackage{\dots}`;
- le definizioni dei comandi e degli ambienti personalizzati (che si impostano con i comandi `\newcommand` e `\newenvironment`);
- la scelta delle varie opzioni generali;
- informazioni come `\title{\dots}`, `\author{\dots}`, eccetera.

Fra `\begin{document}` ed `\end{document}` va il testo vero e proprio, o *corpo del documento*.

La figura 4.1 nella pagina seguente mostra un esempio di documento di piccola lunghezza, composto con $\text{\LaTeX}$. Il documento è stato ottenuto compilando (due volte) il codice seguente:

```
\documentclass[a4paper]{article}  
\usepackage[italian]{babel}  
\usepackage[latin1]{inputenc}  
\usepackage[T1]{fontenc}  
\author{L.~Pantieri}  
\title{Minimalismo}  
\begin{document}  
\maketitle  
\tableofcontents  
\section{Inizio}  
Bene, qui inizia il mio grazioso articolo.  
\section{Fine}  
\dots e qui finisce.  
\end{document}
```

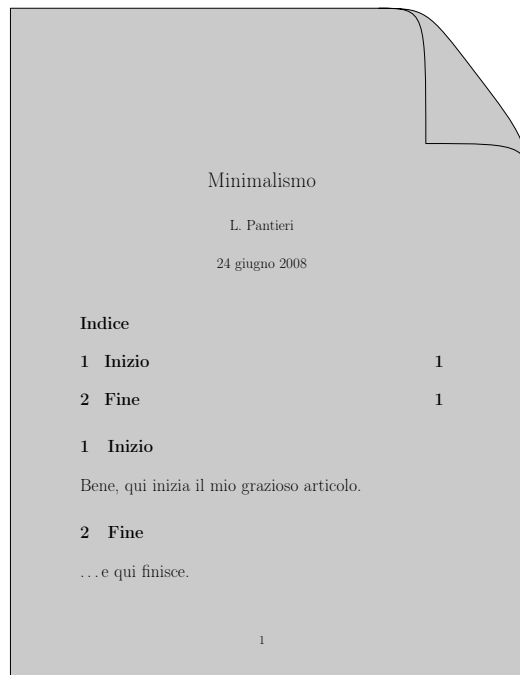


Figura 4.1: Un breve esempio di documento composto con $\text{\LaTeX}$.

Con `\begin{document}` inizia il corpo del documento. Il comando `\maketitle` produce il titolo. Il comando `\tableofcontents` produce l'indice generale: alla prima compilazione, $\text{\LaTeX}$ prende annotazioni in file ausiliari `.aux` e `.toc`, che vengono inserite con la seconda compilazione. Il comando `\section{<titolo del paragrafo>}` inizia un paragrafo. Il comando `\dots` produce i punti ellittici "...". Il comando `\end{document}` termina il documento.

4.4 Le classi di documento

La prima informazione che $\text{\LaTeX}$ richiede quando elabora un file sorgente è il tipo di documento che l'autore desidera realizzare; questo si specifica con il comando `\documentclass`:

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

Di seguito vengono elencate le principali classi standard di documento.

`article` si usa per scrivere articoli;

`report` serve per comporre relazioni contenenti diversi capitoli ed eventualmente dotate di un sommario;

`book` si usa per scrivere libri o tesi;

`letter` permette di scrivere lettere.

Esistono numerose altre classi aggiuntive per comporre svariati documenti, compresi lucidi e presentazioni.³

Il parametro *<opzioni>* del comando `\documentclass` adatta il comportamento della classe del documento; le opzioni devono essere separate da virgole. Di seguito vengono descritte le opzioni globali più comuni per le tre classi principali `article`, `report` e `book` (dal momento che la classe `letter` è fondamentalmente diversa dalle altre tre, per essa si rimanda alla relativa documentazione).

`10pt`, `11pt`, `12pt` Queste opzioni impostano la dimensione del font principale nel documento. Il valore predefinito è di 10 punti.

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

`oneside`, `twoside` Specificano se deve essere generato un documento a doppia o singola facciata. Le classi `article` e `report` sono a singola facciata e la classe `book` è a doppia facciata per impostazione predefinita.

`openany`, `openright` L'opzione `openright`, predefinita nella classe `book`, fa iniziare i capitoli sempre in una pagina destra, mentre l'opzione `openany`, predefinita nella classe `report`, li fa iniziare nella successiva pagina a disposizione. Queste opzioni non sono disponibili con la classe `article`, che non ha capitoli.

`twocolumn` Dà istruzioni a $\text{\LaTeX}$ di impaginare il documento in due colonne.

`titlepage`, `notitlepage` Queste opzioni specificano se dopo il titolo del documento debba avere inizio o no una nuova pagina. La classe `article` non dà inizio a una nuova pagina per impostazione predefinita, al contrario delle classi `report` e `book`.

`fleqn` Compone le formule visualizzate allineandole a sinistra (rispetto a un margine rientrato) invece che centrandole.

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

`draft`, `final` L'opzione `draft` evidenzia le righe che $\text{\LaTeX}$ non è riuscito a comporre adeguatamente, e che quindi fuoriescono dal margine, con una spessa linea nera sul margine destro, rendendole più facili da individuare. Ciò non accade con l'opzione `final`. Le due opzioni sono caricate anche dagli altri pacchetti e hanno effetto sul loro funzionamento. Per esempio, il pacchetto `graphicx` non carica le immagini quando è specificata l'opzione `draft`: al loro posto viene mostrato un riquadro, di dimensioni corrispondenti, che contiene solo il nome dell'immagine.

Per esempio, un file sorgente per un documento composto con $\text{\LaTeX}$ potrebbe iniziare con la riga

³Tra le classi non standard più diffuse vi sono `memoir` e `KOMA-Script` (che consentono di personalizzare qualunque aspetto del documento in modo particolarmente versatile), `toptesi` (specifica per tesi di laurea e dottorato), `beamer` (per fare presentazioni), `examdesign` (per scrivere compiti in classe e temi d'esame). La documentazione di queste classi è disponibile su <http://www.ctan.org/>.

Tabella 4.2: Le opzioni più comuni delle classi standard di $\text{\LaTeX}$.

Opzione	book	report	article
10pt	predefinita	predefinita	predefinita
letterpaper	predefinita	predefinita	predefinita
oneside		predefinita	predefinita
twoside	predefinita		
openany		predefinita	<i>indefinita</i>
openright	predefinita		<i>indefinita</i>
titlepage	predefinita	predefinita	
final	predefinita	predefinita	predefinita

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

che ordina a $\text{\LaTeX}$ a impaginare il documento come articolo, con una dimensione dei font di base pari a undici punti, adeguato alla stampa in due facciate in formato A4.

Questo documento è stato composto usando la classe book:

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

Si noti inoltre che nelle classi standard report e article l'ambiente abstract permette di scrivere il sommario. (I libri generalmente non hanno un sommario, ma eventualmente un apposito capitolo introduttivo. È questo il motivo per cui la classe book non ha alcun ambiente predefinito per il sommario.)

4.5 I pacchetti

Scrivendo dei documenti, capita frequentemente di avere alcune esigenze che $\text{\LaTeX}$ di base non soddisfa. Per esempio, nel linguaggio standard che $\text{\LaTeX}$ usa non è possibile gestire direttamente l'inserimento delle immagini né è possibile sillabare testi scritti in lingue diverse dall'inglese o disporre di un metodo per modificare i margini di pagina in maniera comoda. In questi casi si sfrutta la struttura modulare di $\text{\LaTeX}$, che permette di estenderne le capacità: queste estensioni sono dette *pacchetti*.

4.5.1 Che cosa sono?

Fondamentalmente, un pacchetto è un “file di stile” (con estensione .sty), scritto in linguaggio $\text{\LaTeX}$, contenente una serie di istruzioni che permettono di svolgere le operazioni più disparate.

4.5.2 Come sapere se se ne ha bisogno?

In genere, se per ottenere il risultato sperato si deve faticare troppo, probabilmente qualcuno, trovatosi nella stessa condizione, ha già provveduto a creare un pacchetto per semplificare il lavoro.

Viceversa, può capitare che, compilando un file sorgente, $\text{\LaTeX}$ produca un messaggio di errore del tipo `Can't find file gu i t .sty`: ciò significa che viene caricato

un pacchetto (guit, nell'esempio considerato) che non è presente nella distribuzione di $\text{\LaTeX}$ usata. In questo caso bisogna seguire le istruzioni del paragrafo 4.5.4.

4.5.3 Come scoprire qual è il pacchetto “giusto”?

Questo è l'unico passaggio in cui valgono gusto, abilità personale e un pizzico di fortuna: cercando sul motore di ricerca del sito di GuIT oppure su CTAN o su Sarovar, si trovano preziosi riferimenti a soluzioni per una vastissima gamma di problemi.⁴

Nell'esempio dell'inserimento delle immagini, il pacchetto `graphicx` permette di farlo facilmente (vedi il paragrafo 6.2.4 a pagina 67). Per quanto riguarda la sillabazione di testi scritti in lingue diverse dall'inglese, è necessario caricare e configurare opportunamente il pacchetto `babel` (vedi il paragrafo 5.1.1 a pagina 33). Circa la modifica dei margini di pagina, è consigliabile rivolgersi ai pacchetti `layaureo` o `geometry`, se si usano le classi standard (vedi il paragrafo 5.5.1 a pagina 42).

4.5.4 Come installarli?

Nelle distribuzioni di $\text{\LaTeX}$ più diffuse sono inclusi numerosi pacchetti, sufficienti per gran parte delle esigenze di scrittura, ma può essere necessario installarne altri. Le due maggiori distribuzioni, MiKTeX e $\text{\TeX Live}$ (quest'ultima dalla versione 2008 in poi), permettono di installare e aggiornare i pacchetti tramite appositi programmi (*package manager*). Se però il pacchetto che si desidera installare non fa parte della distribuzione, bisogna procedere a un'installazione manuale.

Installare o aggiornare un pacchetto con MiKTeX

Se si usa il sistema operativo Windows e si utilizza MiKTeX , è possibile installare *al volo* un pacchetto: quando si compila un documento che fa uso di un pacchetto disponibile con la distribuzione ma non ancora installato, MiKTeX lo aggiunge automaticamente, scaricando i dati dall'archivio locale o da CTAN.

In alternativa, è possibile servirsi del programma $\text{MiKTeX Package Manager}$ (vedi la figura 4.2a nella pagina successiva), seguendo questa procedura:

- Si esegue il programma $\text{MiKTeX Package Manager}$ (dal menu Avvio di Windows si seleziona il sottomenu Programmi; da qui si sceglie la voce MiKTeX , da cui si lancia Browse Packages): apparirà una finestra con l'elenco, in ordine alfabetico, dei pacchetti disponibili nell'archivio (*repository*); in corrispondenza di ciascuno di essi viene indicato se esso è installato e ne viene fatta una breve descrizione.
- Si evidenzia il nome dei pacchetti che si desiderano installare.
- Si preme il pulsante “+” che si evidenzierà e si attende che il processo di installazione sia concluso.

⁴<http://www.guit.sssup.it/ricerca/>, <http://www.ctan.org/>, <http://texcatalogue.sarovar.org/>.

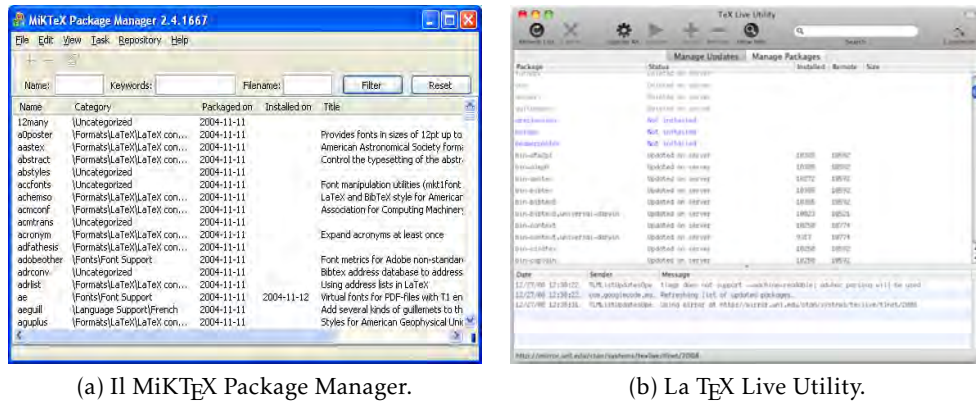


Figura 4.2: La gestione automatica dei pacchetti.

- Si effettua l'aggiornamento dei cosiddetti “file di formato” di MiKTeX (Avvio → Programmi → MiKTeX → Settings → General → Update Formats).

Può capitare che non sia il pacchetto a mancare, ma che la versione che si possiede sia datata. In questo caso si usa il programma MiKTeX Update Wizard (Avvio → Programmi → MiKTeX → Update), che controlla, tramite una semplice interfaccia grafica, quali pacchetti tra quelli installati sono disponibili con una nuova versione e propone di aggiornarli.

Installare o aggiornare un pacchetto con TeX Live

Dal 2008, la distribuzione TeX Live, che su Mac è nota come MacTeX, contiene il programma TeX Live Manager per installare e aggiornare i pacchetti (<http://www.tug.org/texlive/tlmgr.html>). Il programma può essere usato tramite un'interfaccia grafica oppure attraverso la linea di comando. In entrambi i casi il programma va avviato dalla linea di comando, con i privilegi di amministratore (come acquisirli dipende dal sistema operativo su cui si lavora), fornendo, se si desidera l'interfaccia grafica, l'opzione `--gui`. La distribuzione MacTeX fornisce anche un'interfaccia grafica nativa per Mac: TeX Live Utility. Se si usa Mac è quindi preferibile usare quest'ultima, che evita il ricorso a X11 e alla linea di comando.

La TeX Live Utility per Mac Inclusa nell'archivio MacTeXtras (<http://tug.org/mactex/mactextr.html>, vedi la figura 4.2b), la TeX Live Utility è l'interfaccia grafica consigliata agli utenti Mac per installare e aggiornare i pacchetti (richiede Mac OS X 10.5). Quando viene eseguito la prima volta, il programma aggiorna l'installazione di MacTeX. Per aggiornare tutti i pacchetti basta premere il pulsante Update All della barra degli strumenti; si possono aggiornare singoli pacchetti selezionandoli nel relativo elenco e premendo il pulsante Update. I pacchetti si installano usando la scheda Manage Packages.

Gestire i pacchetti dalla linea di comando Se si preferisce usare il programma TeX Live Manager dalla linea di comando (scelta comune fra gli utenti di Linux), queste sono le operazioni più comuni da compiere:

- `tlmgr list` elenca tutti i pacchetti disponibili, contrassegnando con i quelli già installati;
- `tlmgr install <pacchetto>` installa il pacchetto prescelto;
- `tlmgr update --list` elenca i pacchetti aggiornabili;
- `tlmgr update --all` aggiorna tutti i pacchetti;
- `tlmgr update <pacchetto>` aggiorna il pacchetto prescelto.

Per acquisire i privilegi di amministratore, se si usa Linux (o Mac) è necessario premettere a ogni istruzione il comando `sudo` (*super user do*) o, in alternativa, dare le istruzioni in qualità di utente *root* — quale alternativa seguire dipende dal tipo di sistema operativo usato.

L'interfaccia grafica rispecchia fedelmente le operazioni del programma da riga di comando. Dopo averla avviata, con `tlmgr --gui`, bisogna caricare l'elenco dei pacchetti, premendo il tasto *Carica*, nella parte alta della prima schermata. Anche in questo caso, i pacchetti già installati verranno visualizzati con una *i* accanto. Per installare i pacchetti prescelti basta selezionarli e premere il tasto *Installa selezionati*.

Per effettuare l'aggiornamento di uno, più o tutti i pacchetti, bisogna spostarsi nella scheda *Aggiornamento* e usare i tasti *Aggiorna selezionati* (dopo aver effettuato una selezione) oppure *Aggiorna tutti*.

Installare manualmente un pacchetto

Talvolta (ma è un caso che si verifica di rado) può essere necessario installare manualmente un pacchetto non compreso nella propria distribuzione. In questo caso bisogna sapere dove va sistemato il relativo file `.sty`, in modo che $\text{\LaTeX}$ lo possa trovare. Se si usano MiKTeX o $\text{\TeX Live}$, è sufficiente copiare il file `.sty` nella sottocartella `/tex/latex/` dell'*albero personale* (ovvero la cartella in cui il singolo utente può inserire i pacchetti, le classi e gli stili che usa e che non sono inclusi nella distribuzione), eventualmente creando tale sottocartella, se non ci fosse già.⁵

La posizione dell'albero personale dipende dalla particolare distribuzione usata.

- Su Windows, se si usa la distribuzione MiKTeX l'albero personale è la cartella `C:\Documents and Settings\<nome dell'utente>\Dati applicazioni\MiKTeX\2.7\` oppure, equivalentemente, la cartella `C:\Documents and Settings\<nome dell'utente>\Impostazioni locali\Dati applicazioni\MiKTeX\2.7\`.
- Su Mac, l'albero personale è `$HOME/Library/texmf/`.

⁵Le distribuzioni MiKTeX e $\text{\TeX Live}$ prevedono altre due cartelle in cui trovare i file da usare: l'*albero principale* (ovvero la cartella in cui sono contenuti i file installati con la distribuzione usata, che sono disponibili per tutti gli utenti) e l'*albero locale* (ovvero la cartella che rende i file disponibili a tutti gli utenti senza dover modificare manualmente l'albero principale). Per i dettagli: <http://www.guit.sssup.it/installazione/pacchetti.php>.

- Su Linux, la posizione dell'albero personale dipende dalla particolare distribuzione usata. La radice dell'albero personale in qualsiasi distribuzione può essere visualizzata dalla linea di comando con `kpsexpand '$TEXMFHOME'`. Su una Debian, per esempio, l'albero personale è radicato in `~/texmf/`.

In alternativa, è sufficiente copiare i file necessari nella stessa cartella in cui si trova il file `.tex`.

4.5.5 I pacchetti di uso più comune

I seguenti pacchetti di uso comune *devono* essere presenti in tutte le distribuzioni fin dall'inizio. L'unico di questi che richiede un intervento dell'utente è `babel`, per caricare in memoria le regole di sillabazione delle lingue (vedi il paragrafo 5.1.1 a pagina 33).

`amsmath` fornisce svariate estensioni per il miglioramento della gestione di documenti che contengono formule matematiche.

`amssymb` arricchisce la scelta di simboli matematici.

`amsthm` migliora la gestione degli enunciati matematici.

`babel` permette l'uso di lingue diverse dall'inglese.

`booktabs` migliora l'aspetto delle tabelle.

`caption` consente di personalizzare le didascalie.

`chngpage` permette di modificare i margini di una singola pagina.

`eurosym` introduce il simbolo dell'euro.

`float` permette di creare oggetti mobili personalizzati e di forzare il posizionamento di un oggetto.

`fontenc` specifica la codifica dei font da usare nel documento stampato.

`geometry` imposta i margini di pagina nelle classi standard.

`graphicx` gestisce l'inserimento delle immagini.

`hyperref` abilita i riferimenti ipertestuali.

`indentfirst` rientra il primo capoverso di ogni unità di sezionamento (rispettando una consuetudine spesso seguita nei documenti italiani).

`inputenc` permette l'inserimento di caratteri accentati da tastiera nel documento sorgente.

`listings` consente l'inserimento di codici, permettendo un controllo molto sofisticato del loro formato.

`makeidx` fornisce comandi per realizzare l'indice analitico.

`microtype` migliora il riempimento delle righe.

`multicol` dispone il testo su più colonne con bilanciamento.

`showlabels` permette di controllare, in fase di revisione, l'uso dei riferimenti `\label`, `\ref`, `\cite`, eccetera.

`subfig` permette di affiancare figure e tabelle.

`tabularx` compone tabelle di larghezza impostata dall'utente.

`url` imposta l'inserimento degli indirizzi Internet.

`varioref` rende più flessibile la gestione dei riferimenti incrociati.

`xcolor` consente la gestione del colore.

4.6 Gli stili di pagina

Le classi standard di $\text{\LaTeX}$ accettano tre combinazioni predefinite di testatina/piè di pagina (i cosiddetti *stili di pagina*). L'argomento $\langle\textit{stile}\rangle$ del comando `\pagestyle{\langle\textit{stile}\rangle}`

stabilisce quale stile sarà usato. Gli stili di pagina predefiniti delle classi standard sono i seguenti.

plain Stampa i numeri di pagina al piè di pagina, lasciando vuota la testatina. Questo è lo stile di pagina predefinito nelle classi `article` e `report`.

empty Imposta la testatina e il piè di pagina in modo che non vi sia stampato nulla.

headings Il piè di pagina è vuoto, mentre la testatina contiene il numero della pagina vicino al margine esterno, seguito dal titolo del capitolo corrente nella testatina di sinistra e preceduto dal titolo del paragrafo corrente nella testatina di destra. Questo vale per le classi `book` e `report`, mentre per la classe `article` si tratta dei titoli del paragrafo e del sottoparagrafo corrente, rispettivamente. Quando si compone con l'opzione `oneside` (predefinita con `report` e `article`), nella testatina compare solo il titolo della suddivisione maggiore. Questo documento è stato composto con lo stile `headings`, predefinito nella classe `book`.

myheadings È simile a `headings`, con la differenza che il contenuto delle testatine deve essere fissato manualmente dal compositore specificando di volta in volta entrambe le testatine oppure solo quella di destra attraverso i comandi `\markboth` oppure `\markright`, rispettivamente.

È possibile cambiare lo stile della pagina corrente con il comando

```
\thispagestyle{\langle\textit{stile}\rangle}
```

È anche possibile creare stili di pagina personalizzati con il pacchetto `fancyhdr`.

4.7 I file con cui si ha a che fare

Quando si lavora con $\text{\LaTeX}$, si ha a che fare con svariati tipi di file. Di seguito è riportato un elenco (non esaustivo) delle relative estensioni.

4.7.1 I file dell'utente

- `.tex` È l'estensione del testo sorgente scritto con $\text{\LaTeX}$ dall'utente; può essere diviso in più file.
- `.pdf`, `.jpg`, `.png` Sono i formati standard delle figure da inserire in un documento con $\text{\PDFLaTeX}$ (vedi il capitolo 6).
- `.eps` È il formato delle figure da inserire in un documento con $\text{\LaTeX}$ (vedi il capitolo 6).

4.7.2 File di classi, pacchetti e stili

- `.cls` I file delle classi definiscono il tipo di documento; sono scelti con il comando `\documentclass`.
- `.sty` Pacchetto di $\text{\LaTeX}$. È un file che si può caricare nel documento sorgente mediante il comando `\usepackage`.
- `.ins`, `.dtx` File di installazione e documentazione di un pacchetto. Eseguendo $\text{\LaTeX}$ sul file `.ins` e `.dtx` si ottengono rispettivamente il corrispondente file `.sty` e il relativo file `.pdf`.

4.7.3 I file ausiliari

Prodotti dalla compilazione

- `.log` Fornisce un resoconto dettagliato di ciò che è avvenuto durante l'ultima esecuzione del compilatore.
- `.toc` Contiene le informazioni necessarie per produrre l'indice generale (*table of contents*). Viene scritto da $\text{\LaTeX}$ nella prima compilazione ed è letto nella successiva esecuzione del compilatore.
- `.lof` È simile al `.toc` ma riguarda l'elenco delle figure (*list of figures*).
- `.lot` Analogo di `.lof` per l'elenco delle tabelle (*list of tables*).
- `.aux` Un altro file che trasporta informazioni da un'esecuzione all'altra del compilatore; tra le altre cose, il file `.aux` è usato per conservare le informazioni associate ai riferimenti incrociati.

Generati in compilazione da pacchetti e programmi

- .idx È presente se il documento contiene un indice analitico. Tutte le parole che vanno nell'indice analitico sono memorizzate da $\text{\LaTeX}$ in questo file, che deve essere successivamente elaborato con il programma MakeIndex.
- .ind È il risultato dell'elaborazione del file .idx fatta da MakeIndex, pronto per essere incluso nel documento al successivo ciclo di compilazione.
- .ilg Fornisce un resoconto su ciò che è stato compiuto da MakeIndex.
- .out Raccoglie le informazioni per generare i segnalibri ipertestuali, se si usa il pacchetto hyperref (vedi il capitolo 5).

4.7.4 I file di output

- .pdf È il principale risultato dell'esecuzione di $\text{\PDF\LaTeX}$: è il formato più versatile per la stampa e la distribuzione elettronica. Si visualizza e stampa con Adobe Reader o un'applicazione analoga.
- .dvi È il principale risultato dell'esecuzione di $\text{\LaTeX}$. Si può visualizzare e stampare con YAP o un'applicazione analoga.

4.8 Documenti di grandi dimensioni

Un fattore molto importante per il successo nella scrittura di un documento di grandi dimensioni (come un libro o una tesi) è l'organizzazione del proprio materiale. Si suggerisce di collocare tutti i file del documento in una cartella, strutturandola come segue:

- La cartella Capitoli contiene il materiale principale, suddiviso in capitoli, come Introduzione.tex, StoriaFilosofia.tex, ecc.
- La cartella MaterialeInizialeFinale contiene il materiale iniziale e finale, come i ringraziamenti, la dedica, la bibliografia, ecc.
- La cartella Immagini contiene tutte le figure usate nel lavoro. Se si hanno molte figure, potrebbe essere conveniente organizzare la cartella in sottocartelle, contenenti ciascuna le figure di un capitolo.
- $\langle \text{nome del documento} \rangle$.tex: il file principale del documento, che richiama tutti gli altri.

Questo piccolo accorgimento permette di semplificare notevolmente il proprio lavoro.

Tra $\text{\begin{document}}$ e $\text{\end{document}}$ si caricano i file del proprio documento con

```
\input{\file1}
\input{\file2}
...
```

Per esempio, i capitoli si caricano con

```
\input{Capitoli/Introduzione}  
\input{Capitoli/StoriaFilosofia}  
...
```

Il comando `\input` include semplicemente il file specificato: niente di più, niente di meno. Permette l'annidamento, ovvero è possibile richiamare un file che ne richiama un altro.

Per evitare problemi, è opportuno che il percorso (*path*) dei file inclusi mediante il comando `\input` non contenga spazi.

Per far sì che $\text{\LaTeX}$ esegua un veloce controllo sintattico del documento si usa il pacchetto `syntonly`, che fa sì che $\text{\LaTeX}$ scorra il documento controllando solo la sintassi e l'uso corretto dei comandi, senza fornire alcun output. Poiché $\text{\LaTeX}$ viene eseguito più velocemente in questa modalità, si può risparmiare tempo; l'uso è molto semplice:

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

Quando si vuole realizzare il documento vero e proprio, basta commentare la seconda riga (facendola precedere da un simbolo di percentuale).

Capitolo 5

Il testo

In questo capitolo, che estende le conoscenze basilari su $\text{\LaTeX}$ acquisite nel capitolo precedente, vengono fornite le nozioni necessarie per realizzare veri e propri documenti testuali.

5.1 $\text{\LaTeX}$ multilingue e multiplatforma

5.1.1 Il pacchetto babel

Se si vogliono scrivere documenti in lingue diverse dall'inglese, ci sono due ambiti per cui $\text{\LaTeX}$ deve essere configurato:

1. tutte le voci generate automaticamente (“Indice”, “Elenco delle figure”, “Elenco delle tabelle”, “Capitolo”, ...) devono essere adattate alla nuova lingua;
2. è necessario che $\text{\LaTeX}$ conosca le regole di sillabazione per la nuova lingua.

Queste modifiche si possono compiere caricando il pacchetto babel con l'opzione che specifica la lingua (o le lingue) in cui si scrive:

```
\usepackage[⟨lingua⟩]{babel}
```

Le lingue che il pacchetto consente di usare sono elencate su [11].

Per adattare alla lingua italiana tutte le voci generate automaticamente e per attivare la sillabazione italiana, è necessario caricare il pacchetto babel con l'opzione per l'italiano per mezzo del comando

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

L'uso di babel è *necessario* per avere la sillabazione italiana ma in generale *non sufficiente*, in quanto serve anche che il file che contiene le relative regole di sillabazione sia attivato.

- Se si usa MiKTeX , per attivare la sillabazione italiana è necessario selezionare, dal pannello Languages del menu Settings di MiKTeX , la voce *italian* e aggiornare il sistema (Update Formats).¹

¹Se si usa la versione di base (*basic*) di MiKTeX , è necessario anche selezionare, dal pannello Packages del menu Settings di MiKTeX , la voce Language Support → Italian; dopo di ciò si effettua l'aggiornamento dei “file di formato” e della base di dati di MiKTeX (Update Formats e Refresh FNDB).

- Se si usa T_EX Live su Linux, a seconda della distribuzione può essere necessario lanciare `texconfig-sys` dalla linea di comando (occorrono i privilegi di amministratore), scegliere `hyphenation` e togliere il `%` davanti a `italian` nel file che comparirà (dalla versione 2008 in poi, questa procedura non è più necessaria).
- Se si usa la distribuzione T_EX Live su Mac, non c'è bisogno di fare nulla (tutto l'occorrente viene abilitato automaticamente durante l'installazione).

Per alcune lingue, `babel` definisce anche nuovi comandi che semplificano l'immissione di caratteri speciali. La lingua tedesca, per esempio, contiene molte dieresi (ä, ö, ü); con `babel`, si può immettere una ö battendo "o invece di \o. Alcuni comandi predisposti da `babel` per la lingua italiana sono presentati nel paragrafo 5.7.1 a pagina 45.

Scegliere la lingua

Ogni lingua di un documento multilingue si può sillabare a sé. Per esempio, se si carica `babel` con

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

l'ultima lingua (l'italiano) è quella predefinita. Si passa da una lingua all'altra con

```
\selectlanguage{<lingua>}
```

Le precedenti dichiarazioni fanno cambiare la lingua in modo *globale* (comprese le voci generate automaticamente): per esempio, dopo `\selectlanguage{english}`, `\chapter` produrrà "Chapter...".

```
\begin{abstract}
Versione del sommario in
italiano. \dots
\end{abstract}
```

```
\selectlanguage{english}
\begin{abstract}
English version of the
abstract. \dots
\end{abstract}
\selectlanguage{italian}
```

Sommario

Versione del sommario in
italiano. ...

Abstract

English version of the ab-
stract. ...

In alternativa, è possibile usare il comando

```
\foreignlanguage{<lingua>}{<testo>}
```

Il testo dato come secondo argomento di `\foreignlanguage` viene composto secondo le regole tipografiche di `<lingua>`, ma le voci generate automaticamente non cambiano rispetto alla lingua in vigore. Per esempio, dopo `\foreignlanguage{english}`, `\chapter` produrrà "Capitolo...".

Tabella 5.1: Le principali codifiche di input (la codifica `utf8x` richiede il pacchetto `ucs`, da caricare prima di `inputenc`).

Codifica	Opzione di <code>inputenc</code>
<code>iso-8859-1</code>	<code>latin1</code>
<code>iso-8859-15</code>	<code>latin9</code>
<code>UTF-8</code>	<code>utf8, utf8x</code>
Codepage 1252 (Windows)	<code>ansinew</code>
MacRoman (Mac OS)	<code>applemac</code>
Codepage 850 (OS/2, MS-DOS)	<code>cp850</code>

5.1.2 Il pacchetto `inputenc`

A causa della vocazione multiplatforma e multilingue di $\text{\LaTeX}$, è necessario specificare, all'interno del documento, la codifica con cui è stato scritto il codice sorgente `.tex`. Tale codifica, usata dal calcolatore per consentire di introdurre direttamente da tastiera alcuni caratteri particolari (come le vocali accentate *é, ê, ...*), *dipende dalla piattaforma e dall'editor* impiegati. A tal fine si usa il pacchetto `inputenc` (*input encoding*). L'opzione da passare al pacchetto indica la codifica scelta (vedi la tabella 5.1).

L'opzione consigliata è `latin1`, disponibile sulla maggioranza degli editor su piattaforma Windows (tra cui WinEdt e $\text{\TeX}$ nicCenter), Mac (tra cui $\text{\TeX}$ Sshop) e Linux (tra cui Kile):

```
\usepackage[latin1]{inputenc}
```

In alternativa, si può usare anche l'opzione `utf8`. Occorre impostare adeguatamente le preferenze dell'editor usato.

È preferibile usare le codifiche `latin1` o `utf8` in luogo di `ansinew`, `applemac` e `latin9` (o `utf8x`, che richiede il pacchetto `ucs`, da caricare prima di `inputenc`), diffuse rispettivamente su piattaforma Windows, Mac e Linux, dal momento che `latin1` e `utf8` consentono di evitare problemi di compatibilità quando si opera con documenti scambiati tra piattaforme diverse.²

5.2 La codifica dei font

La codifica dei font *del documento finale stampato* è una questione diversa dalla codifica con cui è scritto il codice sorgente. In mancanza di istruzioni specifiche, $\text{\LaTeX}$ impiega i font Computer Modern, che contengono solo i 128 caratteri del vecchio sistema ASCII a 7 bit. Quando sono richiesti caratteri accentati, $\text{\LaTeX}$ li crea combinando un normale carattere con un accento. Benché il risultato sia

²Per trasformare in `latin1` o `utf8` un file codificato diversamente, è possibile usare il programma Charco, multiplatforma e gratuito (<http://www.marblesoftware.com/Charco.html>). Su Linux è disponibile anche l'analogo `iconv` (<http://www.gnu.org/software/libiconv/>), che opera dalla linea di comando.

Su Linux, per capire la codifica usata nella propria distribuzione è possibile eseguire `printenv LANG` dalla linea di comando.

perfetto, questo approccio impedisce il funzionamento della sillabazione automatica all'interno di parole contenenti caratteri accentati.

Fortunatamente, la maggior parte delle distribuzioni recenti di $\text{\LaTeX}$ include una copia dei font $\text{\EC}$, che hanno lo stesso aspetto dei Computer Modern, ma contengono caratteri speciali per la maggior parte delle lettere accentate usate nelle lingue europee. Impiegando questi font si può migliorare la sillabazione in documenti scritti in lingue diverse dall'inglese, ed è quindi decisamente consigliabile farne uso. I font $\text{\EC}$ sono attivati caricando il pacchetto `fontenc`, specificando l'opzione `T1`:

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

La codifica `T1` dei font rappresenta il nuovo standard di codifica dei caratteri di $\text{\LaTeX}$; tale codifica non è ancora la predefinita solo per motivi di compatibilità con le versioni di $\text{\LaTeX}$ precedenti a $\text{\LaTeX 2}_{\epsilon}$.

In effetti, l'impiego di tale codifica richiede l'uso di font con un formato opportuno (vettoriale) per evitare di dar luogo a problemi di resa allo schermo quando si visualizza il documento in formato `PDF` (con programmi come Adobe Reader, Anteprima o Xpdf): in particolare, se non sono installati i relativi font nel formato opportuno, può accadere che i font appaiano sgranati [2, p. 153]. La soluzione di questo problema è fornita, per esempio, dalla collezione di font `CM-Super`, inclusa nelle distribuzioni $\text{\MiKTeX}$ (installazione completa) e $\text{\TeX Live}$, e attivata automaticamente quando si carica il pacchetto `fontenc` con l'opzione `T1`; i font di tale collezione (visualizzabili anche su `DVI`, purché il programma usato a tal fine sia configurato correttamente) hanno la stessa forma dei font $\text{\EC}$, ma presentano molte migliorie in termini di qualità di resa allo schermo. Per maggiori dettagli su questa famiglia si rimanda alla relativa documentazione.

In alternativa ai font `CM-Super`, è possibile usare i font `Latin Modern`, che forniscono un vasto repertorio di caratteri utili per scrivere nelle lingue neolatine e rappresentano un valido sostituto dei Computer Modern:

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

I font `Latin Modern` (che, come i `CM-Super`, sono visualizzabili anche su `DVI`, purché il programma usato sia configurato correttamente) sono inclusi in tutte le distribuzioni recenti di $\text{\LaTeX}$.

5.3 La struttura del testo

Il punto essenziale nella scrittura di un testo è comunicare idee e conoscenze al lettore, che comprenderà meglio il testo se queste informazioni sono ben strutturate, e ne apprezzerà assai meglio la struttura se la forma tipografica rispecchia la costruzione logica del contenuto [18, p. 17].

A differenza degli altri sistemi di composizione, con $\text{\LaTeX}$ è sufficiente indicare la struttura logica del testo, dalla quale il programma ricava la forma tipografica secondo le regole definite nella classe e nei pacchetti caricati.

La più importante unità di testo in $\text{\LaTeX}$ (e in tipografia) è il *capoverso*. Un capoverso è quel tratto di testo che comincia (di regola con un rientro) con una lettera maiuscola e termina con un “punto e a capo”.

Un capoverso viene definito “unità di testo” perché esso è la forma tipografica che deve riflettere un singolo pensiero coerente, o un’idea. Quindi, se si apre un nuovo discorso, si deve cominciare un nuovo capoverso. Quando si è incerti sulla divisione dei capoversi, si pensi al testo come veicolo di idee e pensieri. Se un periodo continua il pensiero precedente, la divisione deve essere eliminata. Se invece un periodo introduce una linea di pensiero completamente nuova, allora occorre introdurla con un nuovo capoverso.

L’unità di testo immediatamente più piccola è il periodo. Nei testi in lingua inglese e, spesso, anche nei testi in lingua italiana, dopo un punto che conclude un periodo c’è più spazio di quanto ce ne sia dopo uno che termina un’abbreviazione. Di regola, $\LaTeX$ capisce quale sia il punto che si intendeva utilizzare. Se però $\LaTeX$ sbaglia, bisogna fornirgli l’indicazione corretta, come è spiegato nel paragrafo seguente.

La strutturazione del testo si estende anche alle parti di un periodo. Le lingue hanno generalmente regole di punteggiatura piuttosto complicate, ma per molte (comprese la lingua italiana e inglese) di solito è sufficiente ricordare che cosa rappresenta la virgola: una breve pausa nel discorso. Se non si è sicuri del posizionamento di una virgola, si legga la frase ad alta voce, prendendo un breve respiro a ogni virgola. Se questo in qualche punto dà la sensazione di essere inopportuno, probabilmente la virgola va eliminata. Se in qualche altro punto si sente la necessità di prendere fiato (o di fare una breve pausa), può essere opportuno inserire una virgola. In ogni caso, è bene ricordare che in italiano la virgola non deve mai essere usata per separare il soggetto dal predicato o quest’ultimo dal complemento oggetto.

Infine, i capoversi in un testo devono essere strutturati logicamente a un livello più alto. Un libro è in genere diviso in capitoli (e talvolta in parti); ogni capitolo è spesso diviso in sezioni gerarchiche che in italiano si chiamano paragrafo, sottoparagrafo, sotto-sottoparagrafo, capoverso e sottocapoverso; in inglese esse si chiamano rispettivamente *section*, *subsection*, *subsubsection*, *paragraph*, *subparagraph*. Si noti la presenza dei falsi amici “paragrafo” e “paragraph”. In ogni caso, l’effetto tipografico di scrivere per esempio

```
\section{La struttura del testo}
```

è a tal punto evidente che l’uso di queste strutture ad alto livello si spiega quasi da sé (vedi comunque il paragrafo 5.6 a pagina 43).

5.4 La composizione dei capoversi

La maggior parte delle persone sottovaluta completamente l’importanza di documenti ben strutturati. Molti ignorano persino qual è il significato della suddivisione in capoversi o, specialmente in $\LaTeX$, cominciano un nuovo capoverso senza rendersene conto. Quest’ultimo errore è particolarmente facile da commettere se nel testo si usano formule [18, p. 17].

Iniziare un nuovo capoverso in $\LaTeX$

Con $\LaTeX$, per iniziare un nuovo capoverso si lascia una riga vuota (in alternativa, si può usare il comando `\par`). Osservando gli esempi che seguono, si cerchi di capire perché a volte si lascia una riga vuota (fine capoverso), e altre volte no. (Se non si comprendono ancora tutti i comandi abbastanza bene per capire questi esempi, si leggano interamente questo capitolo e i primi paragrafi del capitolo 7, e poi si ritorni nuovamente a questo paragrafo.)

<pre>% Esempio 1 \dots quando Einstein propose la definizione \begin{equation} E = mc^2, \end{equation} che è allo stesso tempo la più nota e la meno compresa formula della Fisica.</pre>	<p>...quando Einstein propose la definizione</p> $E = mc^2, \quad (5.1)$ <p>che è allo stesso tempo la più nota e la meno compresa formula della Fisica.</p>
<pre>% Esempio 2 \dots che presenta alcuni vantaggi. La formula \begin{equation} I_{\mathrm{D}} = I_{\mathrm{F}} - I_{\mathrm{R}} \end{equation} costituisce la parte centrale di un modello molto diverso di transistor. \dots</pre>	<p>...che presenta alcuni vantaggi.</p> <p>La formula</p> $I_D = I_F - I_R \quad (5.2)$ <p>costituisce la parte centrale di un modello molto diverso di transistor. ...</p>
<pre>% Esempio 3 \dots da cui segue la legge di Kirchhoff sulle correnti: \begin{equation} \sum_{k=1}^n I_k = 0. \end{equation} La legge di Kirchhoff sulle tensioni può essere ricavata\dots</pre>	<p>...da cui segue la legge di Kirchhoff sulle correnti:</p> $\sum_{k=1}^n I_k = 0. \quad (5.3)$ <p>La legge di Kirchhoff sulle tensioni può essere ricavata...</p>

Terminare un capoverso con una formula fuori corpo (vedi il paragrafo 7.1 a pagina 77), come nel terzo degli esempi proposti, è raro, ma comunque lecito. È invece sempre sconsigliabile iniziare un capoverso con una formula, tanto più se fuori corpo.

Capoversi giustificati

In genere, le righe di un testo hanno tutte la stessa lunghezza. Le necessarie interruzioni di riga e gli spazi tra le parole sono inserite automaticamente da $\LaTeX$,

ottimizzando il contenuto dell'intero capoverso. Inoltre, se necessario, $\LaTeX$ divide le parole che non stanno agevolmente su una riga. Il modo in cui sono composti i capoversi dipende dalla classe di documento. Di regola la prima riga di un capoverso è evidenziata con un rientro, e non c'è spazio aggiuntivo tra i capoversi.³

Interrompere una riga senza iniziare un nuovo capoverso

In casi particolari può essere necessario ordinare a $\LaTeX$ di interrompere una riga. I comandi `\` oppure `\newline` permettono di spezzare una riga e di cominciarne una nuova, *senza iniziare un nuovo capoverso*, e dunque senza tener conto dello spazio fra capoversi e del rientro, come in questo caso.

Per specificare uno spazio aggiuntivo tra due linee dello stesso capoverso si usa il comando `\`[*lunghezza*].

5.4.1 La divisione delle parole in fin di riga

In generale, $\LaTeX$ cerca di produrre sempre le migliori interruzioni di riga possibili; se non riesce a trovare il modo di spezzare le righe secondo i suoi severi criteri, lascia che la riga fuoriesca dal margine destro, avvertendo l'utente con un messaggio di "overfull hbox". Benché $\LaTeX$ avverta l'utente quando ciò accade, le righe a cui si riferisce non sempre sono facili da trovare. Usando l'opzione `draft` del comando `\documentclass`, queste righe saranno evidenziate con una spessa linea nera sul margine destro.

Quasi tutte le parole sono divise correttamente in sillabe da $\LaTeX$, tuttavia esistono casi in cui si usano nomi propri oppure parole rare oppure parole per cui si desidera la divisione etimologica invece di quella fonetica (che, almeno per le parole italiane, viene eseguita benissimo dall'algoritmo di $\LaTeX$). Per esempio si possono avere parole come "macroistruzione" (divisione ordinaria: ma-croi-stru-zio-ne; divisione desiderata: ma-cro-istru-zio-ne) o "nitroidrossilamminico" (divisione ordinaria: ni-troi-dros-si-lam-mi-ni-co; divisione desiderata: nitro-idrossil-amminico).

In questa eventualità, se la sillabazione compiuta da $\LaTeX$ non è soddisfacente, è possibile suggerirla con il comando `\hyphenation` (va posizionato nel preambolo). Si devono scrivere le parole sillabate tra parentesi graffe, separate da uno spazio, come nel seguente esempio:

```
\hyphenation{ma-cro-istru-zio-ne nitro-idrossil-amminico}
```

Il comando precedente può anche essere usato quando si vuole che alcune parole non vengano sillabate. È sufficiente scriverle senza trattini come nel seguente esempio:

```
\hyphenation{FORTRAN ma-cro-istru-zio-ne}
```

³In casi particolari può essere utile inserire uno spazio supplementare fra capoversi. I comandi `\bigskip`, `\medskip` e `\smallskip` inseriscono uno spazio verticale più o meno grande. Sono molto comodi perché la grandezza dello spazio che generano è funzione del font utilizzato. Il comando `\vspace{<lunghezza>}` lascia uno spazio verticale pari a *<lunghezza>*. Se lo spazio deve essere aggiunto in cima o in fondo alla pagina, va usata la versione con asterisco del comando: `\vspace*` invece di `\vspace`.

L'esempio precedente permette che “macroistruzione” sia divisa, come pure “Macroistruzione”, e impedisce del tutto che “FORTRAN”, “Fortran” e “fortran” siano divise. Nell'argomento non sono ammessi caratteri speciali o simboli.

Il comando `\hyphenation` può anche essere impiegato per forzare una sillabazione particolare; se per esempio si vuole che la parola “melograno” sia spezzata tra “melo” e “grano” e non in altri punti, è sufficiente scrivere:

```
\hyphenation{melo-grano}
```

Se la parola in questione compare una sola volta, è possibile suggerirne la sillabazione direttamente nel testo con `\-`. Avremo, per esempio:

Il chimico A.~Angeli scoprì
l'acido nitro\-idrossil\-amminico
nel 1896.

Il chimico A. Angeli scoprì l'acido nitro-
idrossilamminico nel 1896.

Va precisato che gli interventi manuali sulla sillabazione dovrebbero sempre essere fatti nella fase di revisione che precede immediatamente la stampa. Spesso è preferibile riformulare una frase che dà luogo a un errore di “overfull hbox” piuttosto che imporre particolari sillabazioni.

Se si usa l'opzione `italian` di `babel` è utile il comando `/`, che consente di andare a capo dopo la barra: la scrittura `modulazione"/demodulazione` viene composta come “modulazione/demodulazione” all'interno di una riga; in fin di riga, “modulazione/demodulazione”.

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

```
\mbox{<testo>}
```

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

Il numero di fax del nostro
dipartimento è
`\mbox{0547 33 38 98}`.

Il numero di fax del nostro dipartimento
è 0547 33 38 98.

L'argomento `\mbox{\emph{nome del
file}}` deve contenere
il nome del file.

L'argomento *nome del file* deve contenere
il nome del file.

5.4.2 Lo spazio tra le parole

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

Ogni eccezione a queste regole generali deve essere specificata dall'autore. Il comando `\` prima di uno spazio produce uno spazio che ignora l'interpunzione. Un carattere tilde `~` produce uno spazio che ignora l'interpunzione e inoltre impedisce un'interruzione di riga. Si confronti:

Hai studiato le dispense scritte
dal prof.~Beccari? \\
Hai studiato le dispense scritte
dal prof. Beccari?

Hai studiato le dispense scritte dal
prof. Beccari?
Hai studiato le dispense scritte dal prof.
Beccari?

Il concetto è spiegato bene nel
paragrafo~\ref{sec:par}. \\
Il concetto è spiegato bene nel
paragrafo \ref{sec:par}.

Il concetto è spiegato bene nel paragra-
fo 5.4.
Il concetto è spiegato bene nel paragrafo
5.4.

Per avere una spaziatura corretta ed evitare che una riga finisca con un punto o inizi con un numero, si usa lo spazio insecabile: le scritture corrette sono `prof.~Beccari` e `paragrafo~\ref{sec:par}`.

Il comando `\@` davanti a un punto specifica che quel punto conclude una frase anche se si trova dopo una lettera maiuscola. Si confronti:

CEE\@. Poi CE\@. Ora UE\@. \\
CEE. Poi CE. Ora UE.

CEE. Poi CE. Ora UE.
CEE. Poi CE. Ora UE.

La spaziatura corretta si ottiene usando il comando `\@`.⁴

5.4.3 Il pacchetto microtype

Si consiglia di caricare il pacchetto `microtype` (richiede `PDFLATEX`),

```
\usepackage{microtype}
```

che migliora il riempimento delle righe abilitando:

- *l'espansione dei font*, ovvero espandendo i caratteri secondo l'asse orizzontale, per ottenere un riempimento ottimale della riga;
- *la protrusione dei caratteri*, ovvero permettendo che certi caratteri sporgano leggermente a fine riga (tipicamente segni di punteggiatura e trattini).

5.4.4 Il rientro sulla prima riga

Per attivare il rientro sulla prima riga di ogni sezione (capitolo, paragrafo, sottoparagrafo, ...), rispettando una consuetudine spesso seguita nei documenti italiani, è necessario caricare il pacchetto `indentfirst`,

```
\usepackage{indentfirst}
```

in quanto la convenzione anglosassone (predefinita su `LATEX`) non lo prevede.

⁴Lo spazio aggiuntivo dopo un punto può essere disabilitato servendosi del comando `\frenchspacing`, che dice a `LATEX` di *non* inserire dopo un punto più spazio di quanto sia inserito dopo ogni normale carattere. Questa è la regola nella tipografia francese. Se si usa `\frenchspacing`, il comando `\@` non è più necessario.

5.5 Le proporzioni di pagina

5.5.1 Il tormentone dei margini

In tipografia, i margini di pagina hanno il compito di delimitare il corpo del testo, in modo che per il lettore sia comodo da maneggiare, lasciando spazio nella pagina per appoggiare i pollici: questa è la ragione principale, considerato anche il fatto che al lettore i margini interni appaiono duplicati in quanto adiacenti, per cui i margini esterni predefiniti delle classi standard a doppia facciata sono più grandi di quelli interni.

Utilizzando le classi standard di $\text{\LaTeX}$, senza pacchetti particolari o comandi che modificano le proporzioni di pagina, la maggior parte degli utenti europei, che stampa su carta in formato A4, trova che i margini impostati da $\text{\LaTeX}$ siano “troppo ampi” e che in generale il foglio non sia “sufficientemente ben riempito”. Prima di buttarsi nella frenesia dell’«allarghiamo un po’ questa strettissima pagina» è però bene riflettere.

Uno dei punti di forza di $\text{\LaTeX}$ è che consente di disinteressarsi completamente delle questioni tipografiche, per dar modo all’autore di concentrarsi unicamente sulla struttura e sui contenuti del proprio documento. Questo fatto dovrebbe sempre essere tenuto presente: usando uno stile scritto da altri l’utente accetta per buone tutte le impostazioni tipografiche scelte per lui dall’autore dello stile e non è più tenuto a studiare tipografia per mettere a punto l’aspetto delle proprie pubblicazioni. Modificare i margini predefiniti, di conseguenza, comporta l’obbligo di studiare un (bel) po’ di tipografia per ottenere risultati accettabili.

L’esperienza mostra che leggere diventa più difficile se ci sono troppi caratteri per riga: ciò dipende dal fatto che l’occhio si affatica spostandosi dalla fine di una riga all’inizio della successiva (questo è il motivo per cui i giornali vengono stampati su più colonne). Questa esperienza è stata codificata da Bringhurst nella celebre regola che considera come ottimale il valore di circa 66 caratteri per riga, contando anche gli spazi, indipendentemente dal font usato [20, p. 21]. Come ampiezza media di un carattere viene assunto il rapporto tra la lunghezza dell’alfabeto latino minuscolo “abcdefghijklmnopqrstuvwxyz” e il numero di lettere che lo compongono (26).

Bisogna tener conto del fatto che la lunghezza di riga adottata da $\text{\LaTeX}$, di suo, è già superiore al limite stabilito dalla regola di Bringhurst per consentire un maggiore riempimento delle pagine in formato A4, su cui si assume avvenire la stampa (come di norma, in Europa, in ambiente universitario o casalingo) [3, p. 5].

Se si usano le classi standard e si desidera comunque avere una maggior copertura della pagina in formato A4 rispetto a quella predefinita, disinteressandosi al contempo del problema tipografico della definizione delle proporzioni di pagina, è consigliabile caricare il pacchetto `layaureo` (della cui documentazione si raccomanda la lettura), che mette a disposizione dell’utente dimensioni di pagina pronte per l’uso, fornendo anche, con la chiave `binding`, un modo semplice per impostare lo spazio destinato alla rilegatura:

```
\usepackage[binding=5mm]{layaureo}
```

Il pacchetto `layaureo` è di facile uso, ma è anche “rigido”, nel senso che lascia all’utente una scarsa possibilità di personalizzare le dimensioni di pagina. Se

Tabella 5.2: I comandi di sezionamento del documento.

<code>\part</code>	<code>\subsubsection</code>
<code>\chapter</code>	<code>\paragraph</code>
<code>\section</code>	<code>\subparagraph</code>
<code>\subsection</code>	

un utente necessita di proporzioni di pagina differenti da quelle di layaureo, per esempio perché è obbligato a seguire delle indicazioni imposte dalla sua università, è possibile usare il pacchetto `geometry` che è completamente configurabile.

Se, per esempio, si ha l'esigenza di comporre un documento in formato A4 con margini superiore e inferiore di 3 cm, e sinistro e destro (ovvero interno ed esterno, per documenti fronte-retro) di 3,5 cm, lasciando uno spazio per la rilegatura pari a 5 mm, è sufficiente scrivere nel preambolo:

```
\usepackage[a4paper,top=3cm,bottom=3cm,left=3.5cm,right=3.5cm,%
bindingoffset=5mm]{geometry}
```

Nella definizione dei margini di pagina, la modifica di comandi interni di $\text{\LaTeX}$ quali `\textwidth`, `\oddsidemargin`, ... è invece sempre sconsigliabile per molte ragioni [6, 22].

Nel caso che siano necessari degli interventi locali a singole pagine, è possibile usare il pacchetto `chngpage`.

5.5.2 L'interlinea

L'interlinea predefinita di $\text{\LaTeX}$ garantisce un risultato tipografico ottimale e non andrebbe modificata senza una ragione precisa. Tuttavia, a volte, l'editore o i regolamenti di facoltà impongono all'autore un'interlinea differente da uno (predefinita su $\text{\LaTeX}$). Per modificare l'interlinea del documento è consigliabile caricare il pacchetto `setspace`. Tale pacchetto fornisce tre interlinee predefinite richiamate con i comandi `\singlespacing` (interlinea singola), `\onehalfspacing` (interlinea 1,5) e `\doublespacing` (interlinea doppia).

5.6 Il sezionamento del documento

Strutturare un documento (suddividerlo in parti, capitoli, paragrafi, sottoparagrafi, ...) è molto facile da realizzare con $\text{\LaTeX}$. A tal fine, è necessario usare i comandi della tabella 5.2, che prendono come argomento il relativo titolo (spetta all'autore usarli nell'ordine corretto). Il comando `\chapter` è disponibile nelle classi `report` e `book`, ma non nella classe `article`, che non ha capitoli.

Dal momento che la classe `article` non ha capitoli, è molto semplice raggruppare degli articoli come capitoli di un libro.

La spaziatura tra unità di sezionamento, la numerazione e il corpo usato nei titoli sono impostati automaticamente da $\text{\LaTeX}$.

Appendici

Il comando `\appendix` non richiede alcun argomento. Cambia semplicemente la numerazione dei capitoli in lettere (per la classe `article` cambia la numerazione dei paragrafi).

5.6.1 Materiale iniziale, principale e finale

Oltre ai comandi di sezionamento appena illustrati, la classe `book` prevede tre dichiarazioni (`\frontmatter`, `\mainmatter` e `\backmatter`) che controllano il formato del numero di pagina e la numerazione dei capitoli.

Nel materiale iniziale (*frontmatter*) le pagine sono numerate con i numeri romani minuscoli (i, ii, iii, ...) e i capitoli non sono numerati (come se si usasse il comando asteriscato `\chapter*`). Nel materiale principale (*mainmatter*) le pagine sono numerate con numeri arabi (la numerazione riparte da 1) e i capitoli sono numerati con numeri arabi. Nel materiale finale (*backmatter*) le pagine sono numerate come nel materiale principale (la numerazione prosegue da quella del materiale principale), ma i capitoli non sono numerati.

5.6.2 L'indice generale

Il comando

```
\tableofcontents
```

produce l'indice nel punto in cui è inserito. Per ottenere l'indice c'è bisogno di due compilazioni successive: la prima scrive il contenuto dell'indice nel file `.toc` (*table of contents*) e la seconda lo include nel documento.

Varianti asterisco

Di tutti i comandi di sezionamento elencati esiste anche una variante "asterisco". La versione asterisco di un comando è costituita da un asterisco `*` aggiunto alla fine del comando. La variante asterisco dei comandi di sezionamento genera titoli che *non* sono numerati e *non* compaiono nell'indice.

Il titolo qui sopra, per esempio, è stato ottenuto con il comando

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

Se si desidera far apparire il titolo nell'indice, si usa il comando:

```
\addcontentsline{toc}{<livello>}{<titolo>}
```

dove *<livello>* è il nome del livello di sezionamento (che può essere `part`, `chapter`, `section`, `subsection` o `subsubsection`) e *<titolo>* è quanto viene scritto nell'indice. Per esempio, per includere il titolo della presente sezione nell'indice si scrive:

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

Se si sta scrivendo un libro, una tesi o una relazione e si desidera inserire un capitolo non numerato, come per esempio una prefazione o dei ringraziamenti (mentre è consigliabile numerare normalmente l'eventuale introduzione, che è un capitolo come gli altri), è decisamente preferibile che nei capitoli non numerati

non vi sia alcuna testatina. Se si usano le classi standard, a tal fine è sufficiente impostare per essi lo stile di pagina `plain` (vedi il paragrafo 4.6 a pagina 29). Per esempio:

```
\chapter*{Prefazione}
\pagestyle{plain}
...
```

```
\chapter{Introduzione}
\pagestyle{headings}
...
```

Se si desiderano comunque le testatine nei capitoli non numerati, va tenuto presente che nelle classi standard il comando `\chapter*` richiede che i contenuti delle testatine ottenute con lo stile `headings` vengano forniti manualmente. A tal fine si usa il comando `\markboth`:

```
\markboth{\MakeUppercase{\langle testatina di sinistra\rangle}}%
          {\MakeUppercase{\langle testatina di destra\rangle}}
```

Per esempio:

```
\chapter*{Prefazione}
\markboth{\MakeUppercase{Prefazione}}{\MakeUppercase{Prefazione}}
```

Questo problema non si presenta se i capitoli non numerati vengono inseriti usando l'apposita dichiarazione `\frontmatter`, messa a disposizione dalla classe `book` (i capitoli non numerati vanno inseriti con il comando `\chapter`, senza asterisco, dopo `\frontmatter` e prima di `\mainmatter`).

Specificare le voci per l'indice

Normalmente i titoli appaiono nell'indice esattamente come sono nel testo. Se si desidera evitare ciò (per esempio perché un titolo è troppo lungo per trovarvi posto), si può specificare la voce per l'indice come parametro opzionale prima del vero titolo.

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

5.7 Caratteri speciali e simboli

5.7.1 Virgolette, tratti e punti ellittici

Virgolette

In tipografia esistono tre tipi di virgolette: le *basse* (« »), dette anche «francesi», «caporali» o «sergenti»), le *alte* (“ ”), dette anche “inglesi”) e gli *apici* (‘ ’). In $\text{\LaTeX}$, tutti e tre i tipi di virgolette possono essere inseriti direttamente nel sorgente, purché si usi il pacchetto `inputenc` (con la codifica appropriata). Per inserire le virgolette alte *non* bisogna usare il carattere " come invece si farebbe su una macchina per scrivere (che impiega lo stesso carattere per le virgolette di apertura e di chiusura), ma si devono usare gli appositi simboli “ ”. In alternativa, si possono scrivere

due ` (il paragrafo 4.2.2 a pagina 18 spiega come farlo con la tastiera italiana) per le virgolette aperte e due ' (due normali apostrofi nella nostra tastiera) per le virgolette chiuse. Ciò vale anche per inserire gli apici, ovviamente scrivendo solo una volta i caratteri ` e '.

La Delta di Dirac è una
`funzione impropria`.

La Delta di Dirac è una “funzione
impropria”.

Se si usa l'opzione `italian` di `babel` le virgolette alte aperte possono anche essere inserite con `"` (ovvero due volte le normali virgolette nella tastiera italiana), mentre si può scrivere `<` per ottenere i caporali aperti e `>` per i caporali chiusi.

Tratti

In $\text{\LaTeX}$, si distinguono quattro tipi di tratto. Tre di questi si ottengono con un diverso numero di trattini consecutivi. Il quarto è il segno matematico meno:

Stratford-on-Avon, e-mail \\
pag.~13--67 \\
sì~---~oppure no? \\
\$0\$, \$1\$ e \$-1\$

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

I loro nomi sono: “-” trattino breve (*hyphen*), “–” trattino medio (*en-dash*), “—” trattino lungo (*em-dash*) e “−” meno.

Punti ellittici

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

Non così ... ma così: \\
Londra, Parigi, Berlino, \dots

Non così ... ma così:
Londra, Parigi, Berlino, ...

Se, in una citazione, si usa una parte di periodo, è necessario scrivere il segno di *omissione* [...]. A tal fine, è opportuno definire nel preambolo un apposito comando `\omissis`

```
\newcommand{\omissis}{[\ldots\negthinspace]}
```

e usarlo sempre (ricordando che gli spazi dopo i comandi come questo sono ignorati; se occorre uno spazio, bisogna scrivere `\omissis{ }`).

5.7.2 Gli indirizzi Internet e i riferimenti ipertestuali

Per scrivere un indirizzo Internet è utile il pacchetto `url` (se si usa `hyperref`, questo pacchetto è caricato automaticamente):

```
\url{http://profs.sci.univr.%  
it/~gregorio/}
```

[http://profs.sci.univr.it/
~gregorio/](http://profs.sci.univr.it/~gregorio/)

Tabella 5.3: Loghi particolari (il logo $\mathcal{A}\mathcal{M}\mathcal{S}$ richiede il pacchetto `amsmath`; i loghi $\mathcal{G}\mathcal{U}\mathcal{I}\mathcal{T}$ e $\mathcal{A}\mathcal{r}\mathcal{s}\mathcal{T}\mathcal{E}\mathcal{X}\mathcal{n}\mathcal{i}\mathcal{c}\mathcal{a}$ richiedono il pacchetto `guit`).

Comando	Risultato	Descrizione
<code>\TeX</code>	$\mathrm{T}\mathrm{E}\mathrm{X}$	Il “motore” di $\mathrm{L}\mathrm{A}\mathrm{T}\mathrm{E}\mathrm{X}$
<code>\LaTeX</code>	$\mathrm{L}\mathrm{A}\mathrm{T}\mathrm{E}\mathrm{X}$	Il nome del gioco
<code>\LaTeXe</code>	$\mathrm{L}\mathrm{A}\mathrm{T}\mathrm{E}\mathrm{X} 2_{\epsilon}$	L’attuale versione di $\mathrm{L}\mathrm{A}\mathrm{T}\mathrm{E}\mathrm{X}$
<code>\AmS</code>	$\mathcal{A}\mathcal{M}\mathcal{S}$	American Mathematical Society
<code>\GuIT</code>	$\mathcal{G}\mathcal{U}\mathcal{I}\mathcal{T}$	Gruppo Utilizzatori Italiani di $\mathrm{T}\mathrm{E}\mathrm{X}$ e $\mathrm{L}\mathrm{A}\mathrm{T}\mathrm{E}\mathrm{X}$
<code>\Ars</code>	$\mathcal{A}\mathcal{r}\mathcal{s}\mathcal{T}\mathcal{E}\mathcal{X}\mathcal{n}\mathcal{i}\mathcal{c}\mathcal{a}$	La rivista del $\mathcal{G}\mathcal{U}\mathcal{I}\mathcal{T}$

Per abilitare i riferimenti ipertestuali all’interno del documento si usa il pacchetto `hyperref` (di regola, va caricato per ultimo):

```
\usepackage{hyperref}
```

Per sfruttare appieno le potenzialità del pacchetto, è necessario compilare il documento con $\mathrm{PDF}\mathrm{L}\mathrm{A}\mathrm{T}\mathrm{E}\mathrm{X}$ (vedi il paragrafo 3.6 a pagina 13).

Per realizzare un collegamento a un indirizzo di posta elettronica, conviene definire nel preambolo un apposito comando `\mail`,

```
\newcommand{\mail}[1]{\href{mailto:#1}{\texttt{#1}}}
```

da usare come segue:

```
\mail{sit6113@iperbole.%sit6113@iperbole.bologna.it}
bologna.it}
```

5.7.3 Loghi, accenti e caratteri speciali

Loghi

La tabella 5.3 riporta alcuni semplici comandi per comporre particolari loghi.

Accenti e caratteri speciali

Con $\mathrm{L}\mathrm{A}\mathrm{T}\mathrm{E}\mathrm{X}$ è consentito l’uso degli accenti e dei caratteri speciali usati in molte lingue. La tabella 5.4 nella pagina successiva mostra, fra l’altro, ogni sorta di accento applicato alla lettera “o”. Naturalmente, i comandi funzionano anche con le altre lettere.

```
H\^otel, na\"if, \acute{e}l\`eve,      H\^otel, na\"if, \acute{e}l\`eve, Sch\"onbrunner, ¡Se\~norita!.
Sch\"onbrunner, !`Se\~norita!.
```

Si consiglia di racchiudere i caratteri speciali tra parentesi graffe:

```
Sm{\o}rrebr{\o}d,      Smørrebrød, Schloß, Straße.
Schlo{\ss}, Stra{\ss}e.
```

Il segno dell’euro (€) si ottiene con il comando `\euro`, messo a disposizione dal pacchetto `eurosym`.

Tabella 5.4: Accenti e caratteri speciali.

ò	\=o	ó	\'o	ô	\^o	õ	\~o
ō	\=o	ô	\.o	ö	\"o	ç	\c{c}
ö	\u{o}	ö	\v{o}	ő	\H{o}	ø	\c{o}
ø	\d{o}	ø	\b{o}	ö	\t{oo}		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	ı	\l	Ł	\L
ı	\i	ı	\j	ı	!'	ı	?'

5.8 Il titolo del documento

Il comando

`\maketitle`

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

`\title{<titolo>}`

`\author{<autore>}`

e, opzionalmente,

`\date{<data>}`

prima di chiamare `\maketitle`. Come argomento di `\author` sono ammessi più nomi separati dal comando `\and`. La data predefinita è quella della compilazione; il comando `\date{}` omette la data dal titolo. Esiste anche il comando

`\thanks{<ringraziamenti>}`

che permette di inserire i `<ringraziamenti>`.

Nelle classi standard, il comando `\thanks`, che va dato *dentro* l'argomento di `\author`, `\title` o `\date`, funziona come il comando per inserire delle note a piè di pagina, solo che il riferimento di queste note non è né un numero né una lettera, ma un simbolo tratto da un elenco che contiene l'asterisco, la spada, la spada doppia, eccetera. Collegato ai nomi degli autori il comando permette, per esempio, di specificarne l'istituzione di appartenenza, mentre dato nel titolo consente, fra l'altro, di associargli il nome dell'ente finanziatore della ricerca.

`\author{Lorenzo Pantieri\thanks{Ringrazio i membri del \GuIT.}}`

Nella figura 4.1 a pagina 22 si può trovare un esempio di alcuni dei comandi sopra citati.

Il titolo generato dal comando `\maketitle`, bisogna ammetterlo, è piuttosto spartano. Se si sta scrivendo una tesi di laurea o di dottorato, per comporre il frontespizio è consigliabile usare il pacchetto frontespizio (vedi la figura 5.1 nella pagina successiva), di E. Gregorio, che è accompagnato da un'esauriente documentazione (in italiano). Il pacchetto fornisce comandi che permettono di specificare l'università e la divisione (facoltà o dipartimento) presso cui si discute la tesi, il



Figura 5.1: Un esempio d’uso del pacchetto frontespizio.

candidato, l’anno accademico, i relatori e i correlatori. Sono inoltre previste diverse opzioni, fra cui la possibilità di comporre il frontespizio in caratteri senza grazie e di inserire il logo dell’università o un’immagine in filigrana.

Altrimenti è possibile creare un frontespizio personalizzato grazie all’ambiente `titlepage` (deve trovarsi subito dopo `\begin{document}`), all’interno del quale si è completamente padroni dell’impaginazione.

5.9 I riferimenti incrociati

Nei documenti si trovano spesso dei riferimenti incrociati a sezioni, figure, tabelle, teoremi e particolari porzioni di testo. $\text{\LaTeX}$ fornisce i seguenti comandi per i riferimenti incrociati

```
\label{<etichetta>}
\ref{<etichetta>}
\pageref{<etichetta>}
```

dove `<etichetta>` è un identificatore scelto dall’utente. Il formato dell’etichetta è libero, ma di solito si usa un’abbreviazione come `sec` (per le sezioni), `fig` (per le figure), `tab` (per le tabelle), `thm` (per i teoremi), ... seguita da `:` e da una parola chiave che caratterizza l’oggetto cui ci si riferisce. Le etichette devono essere *uniche* nel documento. $\text{\LaTeX}$ sostituisce a `\ref` il numero della sezione, figura, tabella o teorema dopo il quale si trova il corrispondente comando `\label`. Il comando `\pageref` stampa il numero di pagina del corrispondente comando `\label` (si noti che questi comandi non sanno a che cosa fanno riferimento: il comando `\label` si

limita a registrare l'ultimo numero generato automaticamente). Proprio come per l'indice, c'è bisogno di due compilazioni successive. Per esempio, se si identifica il presente paragrafo con

```
\section{Riferimenti incrociati}
\label{sec:ref}
```

è possibile riferirsi a esso con

Ecco un riferimento a questo
paragrafo: ``vedi il
paragrafo~\ref{sec:ref}``.

Ecco un riferimento a questo paragrafo:
“vedi il paragrafo 5.9”.

In molti casi, specialmente quando ci sono una o più pagine tra il riferimento e l'oggetto, è utile avere un riferimento completo della pagina. A tal fine è opportuno usare il pacchetto `varioref` che introduce il comando `\vref`, da usarsi nello stesso modo di `\ref`, ma che aggiunge un riferimento addizionale alla pagina, come “nella pagina precedente/successiva” o “a pagina 23” a seconda di dove sia posizionata la corrispondente `\label`.

Vedi il
paragrafo~\vref{sec:titolo}.

Vedi il paragrafo 5.8 a pagina 48.

5.10 Note a margine e a piè di pagina

Note a margine

Una nota a margine si ottiene molto semplicemente con il comando

```
\marginpar{<nota a margine>}
```

Nei documenti fronte-retro, le note sono poste nel margine destro nelle pagine dispari e nel margine sinistro nelle pagine pari. Nei documenti solo-fronte, le note sono sempre poste nel margine destro.

Note a piè di pagina

Il comando

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

stampa una nota in fondo alla pagina corrente. Le note a piè di pagina (di cui è bene non abusare) dovrebbero sempre essere poste alla fine del relativo capoverso, *dopo* il segno di interpunzione.⁵

Le note a piè di pagina sono
l'emblema della
meticolosità. \footnote{Questo
ne è un esempio.}

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

^aQuesto ne è un esempio.

(Se si compone il titolo di un documento con `\maketitle`, le note ai nomi degli autori, al titolo e alla data si impostano con il comando `\thanks`; vedi il paragrafo 5.8 a pagina 48.)

⁵Così.

5.11 Parole evidenziate

Quando si batte a macchina, le parole più importanti sono evidenziate sottolineandole. In tipografia, invece, le parole sono evidenziate stampandole in *corsivo*. $\text{\LaTeX}$ fornisce il comando

```
\emph{⟨testo⟩}
```

per evidenziare il $\langle\text{testo}\rangle$. Ciò che effettivamente il comando fa al suo argomento dipende dal contesto:

```
\emph{Se si evidenzia  
all'interno di un testo  
evidenziato, \LaTeX{} usa  
il \emph{font normale}  
per evidenziare.}
```

*Se si evidenzia all'interno di un testo evi-
denziato, $\text{\LaTeX}$ usa il font normale per
evidenziare.*

È importante osservare la differenza tra i due ruoli logici del corsivo e dell'evidenziato.

```
\emph{Sono qui in  
\textit{dieci} minuti.}  
  
\textit{Sono qui in  
\emph{dieci} minuti.}
```

*Sono qui in dieci minuti.
Sono qui in dieci minuti.*

5.12 Ambienti

Un generico ambiente si invoca con

```
\begin{⟨ambiente⟩}  
...  
\end{⟨ambiente⟩}
```

dove $\langle\text{ambiente}\rangle$ è il nome dell'ambiente. Gli ambienti possono essere chiamati più volte l'uno all'interno dell'altro, purché l'ordine di chiamata venga rispettato.

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

Di seguito vengono illustrati alcuni ambienti notevoli.

5.12.1 Elenchi puntati, numerati e descrizioni

Gli ambienti che realizzano elenchi sono usati molto spesso in $\text{\LaTeX}$. Permettono infatti:

- di “dare respiro” al testo;
- di migliorarne la leggibilità;

- di strutturare le proprie idee.

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

Gli ambienti che realizzano elenchi permettono:

```
\begin{itemize}
  \item di ``dare respiro'' al testo;
  \item di migliorarne la leggibilità;
  \item di strutturare le proprie idee.
\end{itemize}
```

Ogni elemento dell'elenco comincia con il comando `\item`. Si noti come gli elementi all'interno dell'ambiente `itemize` siano rientrati: in questo modo il codice sorgente è più leggibile.

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

<code>\begin{enumerate}</code>	
<code>\item primo elemento;</code>	1. primo elemento;
<code>\item secondo elemento;</code>	2. secondo elemento;
<code>\item terzo elemento.</code>	3. terzo elemento.
<code>\end{enumerate}</code>	

L'ambiente `description` si usa per le descrizioni. L'ambiente permette di scegliere il testo che appare prima di ogni elemento dell'elenco tramite il parametro "opzionale" (in realtà obbligatorio, in questo caso) del comando `\item`:

<code>\begin{description}</code>	
<code>\item[itemize] per fare</code>	itemize per fare elenchi puntati;
<code>elenchi puntati;</code>	
<code>\item[enumerate] per fare</code>	enumerate per fare elenchi numerati;
<code>elenchi numerati;</code>	
<code>\item[description] per fare</code>	description per fare elenchi in cui ogni
<code>elenchi in cui ogni</code>	elemento comincia con un testo a
<code>elemento comincia con</code>	piacere.
<code>un testo a piacere.</code>	
<code>\end{description}</code>	

Si noti che il testo del parametro "opzionale" di `\item` è stato messo automaticamente in evidenza da $\text{\LaTeX}$ (i dettagli tipografici di questo processo dipendono dalla classe del documento).

Si possono annidare gli elenchi, cioè mettere degli elenchi dentro altri elenchi:

È consigliabile usare gli ambienti per gli elenchi perché:

```
\begin{enumerate}
\item sono facili da usare;
\item rendono più chiaro il
      testo:
\begin{itemize}
\item[-] spaziandolo;
\item[-] facilitandone la
      lettura;
\end{itemize}
\item permettono di
      strutturare meglio
      le proprie idee.
\end{enumerate}
```

È consigliabile usare gli ambienti per gli elenchi perché:

1. sono facili da usare;
2. rendono più chiaro il testo:
 - spaziandolo;
 - facilitandone la lettura;
3. permettono di strutturare meglio le proprie idee.

Di seguito sono riportate alcune convenzioni tipografiche per gli elenchi:

- per gli elenchi semplici (i cui elementi contengono una sola frase) ogni elemento deve terminare con un punto e virgola, tranne l'ultimo, che termina con un punto;
- per gli elenchi i cui elementi contengono due o più frasi, ogni elemento termina con un punto.

Per personalizzare gli elenchi è disponibile il pacchetto `enumitem`, che è accompagnato da una dettagliata documentazione.

5.12.2 Centrare e allineare i capoversi

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

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

Questo testo è allineato a sinistra. $\text{\LaTeX}$ non cerca di creare righe di uguale lunghezza.

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

Questo testo è allineato a destra. $\text{\LaTeX}$ non cerca di creare righe di uguale lunghezza.

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

Al centro
della Terra.

5.12.3 Citazioni e versi

Esistono due tipi di citazioni:

Una citazione in corpo (o “in linea”) è scritta in linea con il corpo del testo (“incorporata nel testo”), come per esempio quando si cita il motto kantiano «il cielo stellato sopra di me, la legge morale dentro di me».

Una citazione fuori corpo (o “in display”) è formata da uno o più capoversi staccati dal testo precedente e seguente mediante spazi di ampiezza adeguata.

Le citazioni in corpo vanno inserite fra virgolette del tipo scelto (vedi il paragrafo 5.7 a pagina 45) e devono essere brevi (un paio di frasi al massimo).

Nel caso di citazioni fuori corpo, il corpo dei capoversi citati è di regola minore del corpo del testo: questo significa, in particolare, che le dimensioni del carattere e l’interlinea devono venire ridotte, e i margini aumentati adeguatamente.

Dal momento che gli ambienti predefiniti di $\text{\LaTeX}$ rispondono solo parzialmente alle esigenze tipografiche delle citazioni fuori corpo, conviene definire (nel preambolo) un nuovo ambiente `citazione` in questo modo:

```
\newenvironment{citazione}
  {\begin{quotation}\small}
  {\end{quotation}}
```

che permette di ottenere una citazione come la seguente:

Dal momento che gli ambienti predefiniti di $\text{\LaTeX}$ rispondono solo parzialmente alle esigenze tipografiche delle citazioni fuori corpo, conviene definire un nuovo ambiente `citazione`.

Con l’ambiente così definito è possibile racchiudere citazioni fuori corpo tra i comandi `\begin{citazione}` e `\end{citazione}`.

Se si è scelto di attivare il rientro sulla prima riga di ogni paragrafo, sottoparagrafo, ..., anche nelle citazioni il periodo iniziale va rientrato: ciò risulta automatico se si sta usando il pacchetto `indentfirst`.

Una regola pratica per la lunghezza delle righe è:
`\begin{citazione}`
 Nessuna riga dovrebbe contenere più di 66~caratteri.
`\end{citazione}`
 Ecco perché le pagine in $\text{\LaTeX}$ hanno margini predefiniti così larghi, e nei giornali si usano spesso più colonne.

Una regola pratica per la lunghezza delle righe è:

Nessuna riga dovrebbe contenere più di 66 caratteri.

Ecco perché le pagine in $\text{\LaTeX}$ hanno margini predefiniti così larghi, e nei giornali si usano spesso più colonne.

Le *intercitazioni*, cioè le citazioni dentro un’altra citazione, vanno inserite fra virgolette di tipo diverso; nel caso di citazioni fuori corpo, il problema ovviamente non sussiste.

Per le poesie è disponibile l’ambiente `verse`. Nell’ambiente `verse`:

- i margini sono aumentati come per le citazioni;

- le linee devono finire con `\\` per andare a capo, tranne l'ultima di ogni strofa;
- le strofe sono separate da linee bianche.

La seguente poesia ``La sera''
è di Fabiano Busdraghi.

```
\begin{verse}
Il sole è già svanito \\
dietro le colline scure \\
e i chiarori del giorno \\
s'avvolgono d'ombra.
```

```
Tutto si assopisce: \\
le tracce di nubi \\
il volo di una rondine \\
i miei ulivi grigi\dots
```

```
e la luce \\
si adagia quieta \\
e come saggia \\
sulla campagna \\
silenziosa.
\end{verse}
```

La seguente poesia "La sera" è di Fabiano Busdraghi.

Il sole è già svanito
dietro le colline scure
e i chiarori del giorno
s'avvolgono d'ombra.

Tutto si assopisce:
le tracce di nubi
il volo di una rondine
i miei ulivi grigi...

e la luce
si adagia quieta
e come saggia
sulla campagna
silenziosa.

5.12.4 Codici

Se si desidera inserire codici all'interno di un documento, è possibile servirsi dell'ambiente `verbatim`. Per esempio, se si desidera inserire un codice $\text{\LaTeX}$ all'interno di un documento, è sufficiente scrivere

```
\begin{verbatim}
Nell'ambiente verbatim
i comandi di \LaTeX{}
e i caratteri speciali
(\{\}%$_\&\#^~)
non vengono interpretati.
\end{verbatim}
```

```
Nell'ambiente verbatim
i comandi di \LaTeX{}
e i caratteri speciali
(\{\}%$_\&\#^~)
non vengono interpretati.
```

Il testo racchiuso tra `\begin{verbatim}` e `\end{verbatim}` viene stampato alla lettera, come se fosse battuto a macchina, con tutti gli spazi e le interruzioni di riga, senza che $\text{\LaTeX}$ interpreti i comandi e i caratteri speciali. Per inserire frammenti di codice in linea con il corpo del testo, si usa il comando `\verb!<testo>!`:

Il logo `` $\text{\LaTeX}$ '' si ottiene
con il comando `\verb!\LaTeX!`.

Il logo " $\text{\LaTeX}$ " si ottiene con il comando
`\LaTeX`.

Il simbolo `!` è solo un esempio di carattere delimitatore. Si può usare qualsiasi carattere tranne le lettere o lo spazio. L'ambiente `verbatim` e il comando `\verb` non possono essere usati come argomenti di altri comandi [6].

In alternativa, è possibile usare il pacchetto `listings`, che permette di avere un controllo molto preciso del formato del codice e riconosce un elevato numero di linguaggi di programmazione.

5.13 La revisione finale

Una delle caratteristiche fondamentali di $\text{\LaTeX}$ è che permette ai suoi utenti di concentrarsi sulla struttura logica del contenuto e non sulla forma, gestita automaticamente dal programma. Tuttavia, al momento attuale, nessun software può risolvere automaticamente *tutti* i problemi tipografici ed evitare di lavorare sulla forma definitiva del documento, se si desidera un prodotto di alta qualità. La revisione finale di un lavoro è una fase impegnativa, ma gratificante: risolvere grandi problemi di impaginazione con piccole modifiche è un'arte complicata, ma ricca di soddisfazioni.

In generale, piuttosto che modificare le impostazioni generali di $\text{\LaTeX}$, è conveniente provare a effettuare piccole modifiche al testo, che spesso sono sufficienti per risolvere i problemi e permettono di ottenere un risultato più elegante.

Supponiamo, per esempio, che l'ultimo paragrafo di un capitolo di un libro sfiori di due righe su una pagina a destra. Si avrebbero due pagine consecutive praticamente bianche. Per evitare questo inestetismo, è possibile provare a riformulare un paio di capoversi, in modo da accorciarli di una riga ciascuno.

In alternativa, si possono usare alcuni strumenti messi a disposizione da $\text{\LaTeX}$ per regolare finemente i risultati, in sede di revisione finale. Per esempio, si può provare a dare il comando

```
\looseness=-1
```

per dire a $\text{\LaTeX}$ di cercare di comporre un capoverso usando una riga in meno della sua lunghezza "naturale" (vanno scelti i capoversi che abbiano almeno cinque o sei righe e la cui ultima riga sia breve). Se il capoverso è sufficientemente esteso, $\text{\LaTeX}$ è in grado di comporlo adeguatamente, accorciandolo di una riga. Il comando `\looseness` va dato all'inizio del capoverso da modificare, senza lasciare alcuna riga vuota fra il comando e il testo del capoverso, nel modo seguente:

```
\dots Qui finisce un capoverso.
```

```
\looseness=-1
```

```
Qui ne comincia un altro, che \LaTeX{} cercherà di comporre usando  
una riga in meno della sua lunghezza ``naturale''. \dots
```

Il valore di `\looseness` è un parametro locale relativo al capoverso in cui compare, quindi l'effetto del comando non si propaga. Se il precedente stratagemma funziona per un paio di capoversi, si ottiene l'effetto voluto di "riportare indietro" le due righe di troppo. Se lo stratagemma non dovesse funzionare, si può usare il comando

```
\enlargethispage{\baselineskip}
```

(va dato fra due capoversi) che allunga di una riga, rispetto al normale, la pagina in cui compare. Dando il comando precedente in una *coppia di pagine affiancate* (nei documenti fronte-retro), l'aggiustamento non dà nell'occhio, se il margine inferiore è sufficientemente ampio e se non ci sono note a piè di pagina.

Il comando `\enlargethispage` è da sconsigliare se si stampa su carta di bassa grammatura, perché l'aggiustamento si vedrebbe in trasparenza sfogliando il volume. Su carta di buona qualità il problema non si pone: l'importante è che non si vedano disparità sulla coppia di pagine affiancate.

In alternativa, si può usare la versione asterisco del comando `\enlargethispage`,
,

```
\enlargethispage*{\baselineskip}
```

che prima di allungare la pagina di una riga cerca di stringere il più possibile gli spazi bianchi verticali al fine di non allungare la pagina, ovvero di allungarla il meno possibile.

I trucchi precedenti si possono usare anche al contrario. Alcuni problemi di impaginazione possono essere risolti *allungando* qualche capoverso o *accorciando* una coppia di pagine affiancate. A tal fine si usano il comando

```
\looseness=1
```

che allunga di una riga il relativo capoverso e il comando

```
\enlargethispage{-\baselineskip}
```

che accorcia di una riga la pagina in cui compare.

Vedove e orfane

In tipografia, si usa chiamare “orfana” una riga solitaria in fondo alla pagina (tipicamente, la prima riga di un capoverso) e “vedova” una riga solitaria in cima alla pagina seguente (tipicamente, l’ultima riga di un capoverso). Entrambi questi casi andrebbero evitati, facendo in modo che ci siano almeno due righe di uno stesso capoverso sia in cima che in fondo a ogni pagina. $\text{\LaTeX}$ è già programmato per ottenere questo effetto, ma a volte può essere necessario un intervento “manuale” dell’utente.

Oltre agli strumenti presentati in questo paragrafo, per eliminare una riga “orfana” può essere utile dare il comando `\pagebreak` prima del capoverso che dà problemi. Il comando `\pagebreak` consente di interrompere una pagina, lasciando anche (a differenza di `\newpage`, che comincia semplicemente una nuova pagina) che $\text{\LaTeX}$ cerchi di giustificarla verticalmente. Se c’è abbastanza spazio bianco nella pagina, il risultato che si ottiene è generalmente buono.

Capitolo 6

Tabelle e figure

Le tabelle e le figure sono fra gli oggetti che vengono usati più frequentemente nella composizione dei documenti. Sono anche fra gli argomenti trattati più estesamente dalle guide. In questo capitolo vengono presentati i concetti e gli strumenti fondamentali per comporre e gestire tabelle e figure con $\text{\LaTeX}$.

6.1 Le tabelle

Le tabelle sono uno degli oggetti che vengono usati più spesso nei documenti scientifici. Questo paragrafo, basato su [15], cui si rimanda per ogni approfondimento, spiega come usare $\text{\LaTeX}$ per comporre tabelle, cercando di affrontare gli aspetti essenziali dell'argomento.

Da qui in avanti diamo per scontato che sia caricato il pacchetto `booktabs`.

6.1.1 Regole generali

La composizione delle tabelle dovrebbe fondarsi sulle seguenti regole [7]:

- non usare *mai* linee verticali;
- evitare linee doppie;
- inserire le unità di misura nell'intestazione della tabella (invece che nel corpo);
- non usare virgolette per ripetere il contenuto di celle.

Così vuole la tradizione tipografica, che contrasta con la cattiva abitudine, purtroppo oggi molto diffusa, di comporre le tabelle come se fossero parti di un foglio elettronico. Per capire l'importanza del rispetto delle regole generali si confrontino le tabelle 6.1 nella pagina successiva e 6.2 nella pagina seguente.

6.1.2 L'ambiente `tabular`

L'ambiente `tabular` è il principale strumento di base offerto da $\text{\LaTeX}$ per la creazione di tabelle. Cominciamo con un semplice esempio:

Tabella 6.1: Tabella non composta correttamente secondo le regole generali.

D	P_u	u_u	β	G_f
5 m	269,8 kg	0,000674 m	1,79	0,04089 Pa · m
10 m	421,0 kg	0,001035 m	3,59	"
20 m	640,2 kg	0,001565 m	7,18	"

Tabella 6.2: Tabella composta correttamente secondo le regole generali.

D (m)	P_u (kg)	u_u (m)	β	G_f (Pa · m)
5	269,8	0,000674	1,79	0,04089
10	421,0	0,001035	3,59	0,04089
20	640,2	0,001565	7,18	0,04089

```

\begin{tabular}{lcr}
\toprule
Apple & Mac~OS~X & 10.5 \\
NeXT & NeXTSTEP & 3.3 \\
Be & BeOS & 5.0 \\
\bottomrule
\end{tabular}

```

Apple	Mac OS X	10.5
NeXT	NeXTSTEP	3.3
Be	BeOS	5.0

Abbiamo assegnato all'ambiente `tabular` un argomento (detto anche *preambolo* della tabella) di tre caratteri: ciò significa che la tabella è formata da tre colonne. Più in dettaglio:

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

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

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

All'interno dell'ambiente `tabular`, le righe terminano con il comando `\\` e le colonne sono separate da `&`. Si noti che $\text{\LaTeX}$ non richiede che le colonne (ovvero i simboli `&`) siano allineate, tuttavia è consigliabile che lo siano per migliorare la leggibilità del sorgente.

Le linee orizzontali si ottengono mediante i tre comandi `\toprule`, `\midrule` e `\bottomrule`. In particolare, `\toprule` e `\bottomrule` vanno usati rispettivamente per la prima e l'ultima riga e hanno uno spessore maggiore delle linee per le altre righe ottenute con `\midrule`. Si noti che non c'è `\\` dopo `\toprule`, `\midrule` e `\bottomrule`.¹

Si veda per esempio la seguente tabella:

¹I comandi `\toprule`, `\midrule` e `\bottomrule` del pacchetto `booktabs` sostituiscono il comando `\hline` offerto da $\text{\LaTeX}$, che ha una resa tipografica non del tutto soddisfacente a causa del poco spazio tra le linee orizzontali e il testo delle celle.

<code>\begin{tabular}{1*{2}{c}}</code>			
<code>\toprule</code>			
<code>& Paese & Secolo \\</code>	<table><tr><th>Paese</th><th>Secolo</th></tr></table>	Paese	Secolo
Paese	Secolo		
<code>\midrule</code>			
<code>Tiziano & Italia & XVI \\</code>	Tiziano Italia XVI		
<code>Cézanne & Francia & XIX \\</code>	Cézanne Francia XIX		
<code>Escher & Olanda & XX \\</code>	Escher Olanda XX		
<code>\bottomrule</code>			
<code>\end{tabular}</code>			

Il codice `*{<n>}{<formato>}`, che equivale a `<n>` volte la dichiarazione `<formato>` (qui, dunque, `cc`), è utile per migliorare la leggibilità del sorgente nel caso di tabelle con molte colonne.

6.1.3 Celle su più colonne

Per avere una cella che si estende su più colonne si usa il comando `\multicolumn`

<code>\begin{tabular}{ll}</code>		
<code>\toprule</code>		
<code>\multicolumn{2}{c}%</code>		
<code>{\textbf{Musei}} \\</code>	<table><tr><th>Musei</th></tr></table>	Musei
Musei		
<code>\midrule</code>		
<code>Louvre & Parigi \\</code>	Louvre Parigi	
<code>Prado & Madrid \\</code>	Prado Madrid	
<code>MoMA & New York \\</code>	MoMA New York	
<code>\bottomrule</code>		
<code>\end{tabular}</code>		

Il comando `\multicolumn{<n>}{<formato>}{<testo>}` combina le successive `<n>` colonne in una singola cella della larghezza delle corrispondenti celle, inclusi gli spazi tra colonne. L'argomento `<formato>` deve contenere un simbolo di posizionamento `l`, `r` o `c`.

6.1.4 Il pacchetto tabularx

Il pacchetto `tabularx` modifica la larghezza di certe colonne al fine di coprire la larghezza della tabella definita dall'utente. Le colonne che possono essere dilatate o compresse sono indicate dal comando di allineamento `X`; il testo che viene inserito in queste colonne viene mandato a capo automaticamente e, in mancanza di altre specificazioni, viene giustificato. Il pacchetto `tabularx` richiede il pacchetto `array`. Per esempio, il codice

```
\begin{tabularx}{\columnwidth}{>{\bfseries}lX}
\toprule Forza & Una forza è una grandezza fisica che si manifesta
nell'interazione di due o più corpi materiali, che cambia lo stato
di quiete o di moto dei corpi stessi. \\
\midrule
Momento polare & Il momento polare di una forza rispetto a una
determinata origine è definito come il prodotto vettoriale tra il
vettore posizione (rispetto alla stessa origine) e la forza. \\
```

Tabella 6.3: Tabella ottenuta con il pacchetto `tabularx`.

Forza	Una forza è una grandezza fisica che si manifesta nell'interazione di due o più corpi materiali, che cambia lo stato di quiete o di moto dei corpi stessi.
Momento polare	Il momento polare di una forza rispetto a una determinata origine è definito come il prodotto vettoriale tra il vettore posizione (rispetto alla stessa origine) e la forza.

```
\bottomrule
\end{tabularx}
```

produce la tabella 6.3. In questa tabella l'utente definisce la larghezza dell'intera tabella — in questo caso pari alla larghezza della colonna di composizione `\columnwidth`, che per documenti scritti in un'unica colonna coincide con la larghezza del corpo del testo `\textwidth` (vedi il paragrafo 6.3 a pagina 68) — e il pacchetto calcola in automatico la larghezza della seconda colonna.

6.1.5 Allineare i numeri alla virgola

Per allineare i numeri alla virgola è possibile usare il pacchetto `dcolumn`, che mette a disposizione un nuovo tipo di colonna:

```
D{\langle sep-in \rangle}{\langle sep-out \rangle}{\langle prima.dopo \rangle}
```

Il primo argomento è il carattere usato nel documento `.tex` per indicare la separazione delle cifre decimali (di solito il punto o la virgola), il secondo quello che si vuole nel documento composto (in italiano la convenzione è la virgola, mentre in inglese viene usato il punto), il terzo è il numero di cifre a sinistra (prima) e a destra (dopo) della virgola. I numeri vengono allineati rispetto al separatore e, se il terzo argomento è negativo, il separatore sarà al centro della colonna. Se le colonne hanno dei titoli, è necessario inserirli all'interno di comandi `\multicolumn{1}{c}{...}`. Per esempio, la tabella 6.4 nella pagina successiva può essere ottenuta con il seguente codice:

```
\begin{tabular}{cD{.}{,}{5.4}}
\toprule
Espressione & & \multicolumn{1}{c}{Valore} \\
\midrule
 $\pi$  & & 3.1416 \\
 $\pi^{\pi}$  & & 36.46 \\
 $\pi^{\pi^{\pi}}$  & & 80662.7 \\
\bottomrule
\end{tabular}
```

6.1.6 Le tabelle mobili

Normalmente le tabelle vanno trattate come “oggetti mobili” (vedi il paragrafo 6.4 a pagina 70): se una tabella non sta sulla pagina corrente, è possibile chiedere a $\text{\LaTeX}$ di stamparla su una pagina successiva, riempiendo quella attuale con il

Tabella 6.4: Esempio di tabella con allineamento alla virgola.

Espressione	Valore
π	3,1416
π^π	36,46
π^{π^π}	80662,7

testo successivo. A tal fine l'ambiente `tabular` deve essere inserito nell'ambiente `table` che:

- permette di generare automaticamente l'elenco delle tabelle con il comando `\listoftables`;
- permette di creare la didascalia e di assegnare un numero progressivo alla tabella con il comando `\caption`;
- permette di assegnare alla tabella un'etichetta con il comando `\label` (che deve comparire *dopo* il corrispondente `\caption`) con cui richiamarla nel testo con `\ref` o `\vref`.

Per centrare una tabella mobile è meglio usare il comando `\centering` invece dell'ambiente `center`, poiché quest'ultimo inserisce uno spazio verticale supplementare esagerato (vedi il paragrafo 6.4.1 a pagina 71). Per separare la didascalia dalla tabella (di regola, la didascalia si scrive *sopra* la tabella), se si usano le classi standard è conveniente caricare il pacchetto `caption` e, una volta per tutte, scrivere nel preambolo

```
\captionsetup[table]{position=top}
```

In questo modo, per ottenere una tabella come la 6.4 è sufficiente scrivere il seguente codice:

```
\begin{table}[tb]
\caption{Esempio di tabella con allineamento alla virgola.}
\label{tab:virgola}
\centering
\begin{tabular}{<...>}
...
\end{tabular}
\end{table}
```

In alternativa, per separare la didascalia dalla tabella, è possibile servirsi del comando `\medskip` (va dato dopo ogni `\caption`, prima del relativo `tabular`).

Ridurre la dimensione del font

Per ridurre la dimensione del font all'interno di una tabella è sufficiente mettere il comando della dimensione del font dentro l'ambiente `table`. Per esempio, il codice

```
\begin{table}[tb]\footnotesize
\caption{<...>}\label{<...>}\centering
```

Tabella 6.5: Tabella con font di dimensione inferiore (`footnotesize`) rispetto al resto del testo (`normalsize`).

Forza	Una forza è una grandezza fisica che si manifesta nell'interazione di due o più corpi materiali, che cambia lo stato di quiete o di moto dei corpi stessi.
Momento polare	Il momento polare di una forza rispetto a una determinata origine è definito come il prodotto vettoriale tra il vettore posizione (rispetto alla stessa origine) e la forza.

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

produce la tabella 6.5. Si confronti il risultato con la tabella 6.3 a pagina 62.

Se la tabella non è mobile, ovvero se non è inserita in un ambiente `table`, la dichiarazione della dimensione del font deve essere fatta fuori dall'ambiente `tabular` e poi alla fine deve essere ripristinato il font usato nel testo. Al contrario, quando la dichiarazione è inserita in un ambiente `table`, vale solo all'interno di tale ambiente e quindi non è necessario ripristinare la dimensione normale del font alla fine.

6.2 Le figure

Le figure sono uno degli argomenti trattati più estesamente dalle guide; esistono anche guide interamente dedicate all'argomento [4], cui si rimanda il lettore per ogni approfondimento.

I problemi incontrati dagli utenti di $\text{\LaTeX}$ durante l'inserimento di figure sono generalmente di due tipi. Una parte dei problemi deriva dalle figure in sé, ovvero dai file che si cerca di inserire in un documento (verranno trattati in questo paragrafo), mentre un altro tipo di problemi, totalmente distinto dal precedente, è quello del posizionamento delle figure (verranno trattati nei paragrafi 6.3 a pagina 68 e 6.4 a pagina 70).

6.2.1 Immagini vettoriali e bitmap

Esistono due grandi classi di figure, le immagini vettoriali e le immagini bitmap. Le prime sono descritte da forme e possono essere scalate e deformate senza perdere definizione; sono soprattutto adatte per i grafici e per gli schemi. Le seconde sono matrici di pixel colorati e sono adatte per le fotografie, i disegni e le icone [16, p. 32].

Immagini vettoriali

Nonostante vi siano vari strumenti per creare grafici e schemi con $\text{\LaTeX}$ oppure con qualche sua estensione, la maggior parte degli utenti li considera decisamente

ostici. Pertanto, questa possibilità non sarà più considerata per il resto di questo lavoro.

Un metodo di gran lunga più semplice per inserire grafici o schemi in un documento consiste nel prepararli attraverso un software specifico e di includere l'immagine ottenuta nel documento. Il seguente è un elenco di programmi di grafica vettoriale.

Inkscape è un programma multiplatforma e gratuito particolarmente adatto per disegnare grafici “qualitativi”.

Gnuplot è un programma multiplatforma e gratuito, che permette di realizzare grafici di funzioni matematiche in due e tre dimensioni.

OmniGraffle è un ottimo programma commerciale per Mac, che permette di realizzare schemi e diagrammi di ogni tipo (<http://www.omnigroup.com/applications/omnigraffle/>).

I formati vettoriali più noti e diffusi sono il formato PDF, il formato PS e il suo parente stretto EPS; oggi è diffuso anche il formato SVG, usato specialmente per le applicazioni Web.

Immagini bitmap

I formati di matrici di pixel sono numerosissimi, e vanno dal JPEG, molto diffuso per rappresentare immagini fotografiche, al PNG, adatto per rappresentare disegni e icone, al GIF al TIFF. Questo è solo un piccolo elenco di formati grafici bitmap, perché esistono dei formati specifici per certi apparecchi fotografici digitali o per particolari codici di colore. Alcuni formati sono compressi in una maniera che sfrutta la ridondanza di informazioni delle immagini.

6.2.2 Conversione dei formati

La prima cosa da fare è produrre figure nel formato più adatto per i propri scopi. È inutile registrare una figura in JPEG per poi convertirla in PDF, in quanto la conversione di un'immagine bitmap in PDF include semplicemente il file bitmap in una “cornice” PDF senza migliorare in alcun modo la qualità. È sbagliato anche fare la conversione opposta da file vettoriale a bitmap, perché in questo modo si perdono le informazioni sulla geometria contenuta nella figura e quindi si abbassa la qualità del file.

Ciò premesso, dal momento che, come anticipato nel paragrafo 3.6 a pagina 13, $\text{\LaTeX}$ richiede immagini esclusivamente in formato EPS, mentre $\text{\PDF\LaTeX}$ accetta immagini in formato PDF (se vettoriali) oppure JPG o PNG (se bitmap), può essere necessario convertire un formato in un altro. A tal fine esistono diversi programmi e di seguito se ne elencano alcuni.

Ghostview è un programma gratuito per Unix/Linux che permette, fra l'altro, di convertire file dal formato EPS e PS al formato PDF. Esiste anche un programma simile, GSview, disponibile per Windows e Unix/Linux. Entrambi i programmi si scaricano dal sito <http://pages.cs.wisc.edu/~ghost/> e richiedono Ghostscript (disponibile allo stesso indirizzo Internet).

Gimp è un programma multiplatforma e gratuito che consente, fra l'altro, di visualizzare le immagini bitmap di svariati formati e di registrarle in uno qualunque dei numerosi formati che è in grado di gestire. Non va usato per convertire formati vettoriali in formati a matrici di pixel, limitandosi alle trasformazioni dei formati bitmap. Si scarica da <http://www.gimp.org/>.

Anteprima è un applicativo del sistema operativo Mac OS X; permette, fra l'altro, di visualizzare e convertire immagini di numerosi formati, sia vettoriali sia bitmap. Su Mac, la conversione di un'immagine EPS o PS in PDF viene fatta automaticamente dal sistema operativo: cliccando su un file EPS o PS, esso viene aperto con Anteprima, dal quale si può registrare il file in formato PDF.

6.2.3 Scontornare le immagini

Uno dei parametri più importanti di una figura è l'informazione relativa alle dimensioni del rettangolo circoscritto all'immagine (il cosiddetto "bounding box"). Questo contorno determina la taglia effettiva dell'immagine e serve a $\text{\LaTeX}$ per calcolare lo spazio da riservare alla figura. Idealmente, il contorno dovrebbe essere al limite massimo del contenuto dell'immagine, ma talvolta capita di aver a che fare con figure con grandi bordi bianchi attorno all'immagine effettiva.

Questo porta spesso a grandi confusioni, perché di fatto $\text{\LaTeX}$ sta lasciando alla figura lo spazio corretto, ma visivamente parte di questo spazio è usato per il bordo bianco, quindi la figura appare troppo piccola, non centrata, con eccessivi margini verticali, eccetera.

La prima cosa da verificare è quindi che le dimensioni del rettangolo circoscritto all'immagine che si desidera includere nel documento siano corrette. Per farlo, basta aprire la figura con un programma accessorio (come Adobe Reader o Gimp) e attivare la visualizzazione del contorno. Se le dimensioni del contorno non sono corrette, è necessario intervenire, ma il problema non ha niente a che vedere con $\text{\LaTeX}$. Nel caso si abbiano poche figure con contorni errati, questi si possono correggere a mano. Se però il problema riguarda molti file, bisogna cercare di correggerlo all'origine (per esempio configurando correttamente il programma di grafica usato per produrre le immagini). Di seguito si elencano alcuni programmi per scontornare le immagini.

Gimp consente di scontornare e di correggere immagini bitmap di numerosi formati.

Ghostview e **GSview** permettono di scontornare immagini vettoriali.

Adobe Acrobat è un programma per Windows e Mac che permette di eseguire numerose azioni sui file e sulle immagini in formato PDF. Si possono per esempio estrarre delle pagine contenenti immagini da un file PDF e poi si possono ritagliare solo le immagini che interessano, scontornandole con una comoda interfaccia grafica. L'unico difetto potrebbe essere il costo, visto che il programma è commerciale (<http://www.adobe.com/it/products/acrobatpro/>), mentre Adobe Reader, gratuito, non consente di effettuare modifiche.

Anteprima permette, fra l'altro, di scontornare immagini di numerosi formati, vettoriali e bitmap.

Tabella 6.6: Opzioni del pacchetto graphicx.

<code>width</code>	Ridimensiona l'immagine alla larghezza specificata
<code>height</code>	Ridimensiona l'immagine all'altezza specificata
<code>angle</code>	Ruota l'immagine in senso antiorario
<code>scale</code>	Riassegna le dimensioni dell'immagine

6.2.4 L'inclusione delle immagini

Per inserire le figure è necessario caricare il pacchetto `graphicx`:

```
\usepackage{graphicx}
```

Per inserire un'immagine nel documento, si usa il comando

```
\includegraphics[<opzione>=<valore>,<...>]{<immagine>}
```

Il parametro opzionale permette l'inserimento di un elenco di opzioni, separate da virgole, con i relativi valori. Le opzioni possono essere usate per modificare la larghezza, l'altezza e l'orientamento del grafico incluso. La tabella 6.6 elenca le opzioni più importanti. Gli esempi che seguono dovrebbero rendere più chiaro il procedimento.

È possibile specificare la grandezza che l'immagine deve assumere, in larghezza,

```
\includegraphics[width=%  
\columnwidth]{Mani}
```



o in altezza,

```
\includegraphics[height=%  
0.15\textheight]{Mani}
```



(`\textheight` indica l'altezza del corpo della pagina, senza contare la testatina e il piè di pagina).

Nonostante sia possibile attribuire alle chiavi `width` e `height` un valore espresso in punti, millimetri e in qualsiasi unità tipografica riconosciuta da $\text{\LaTeX}$ (per esempio `width=87mm`), è opportuno esprimere questa dimensione con un valore legato alla geometria della pagina (per esempio `width=0.5\columnwidth`), perché cambiando le dimensioni della pagina la dimensione relativa mantiene le proporzioni,

mentre la dimensione assoluta potrebbe dare luogo a inconvenienti di impaginazione. Si pensi per esempio di voler passare da una composizione a piena pagina a una composizione a più colonne; 87 mm potrebbero andare bene a piena pagina, ma potrebbero essere troppi per una colonna.

Il comando `\includegraphics` permette inoltre di reimpostare le dimensioni dell'immagine:

```
\includegraphics[scale=0.10]%
{Mani}
```



Si può infine ruotare l'immagine in senso antiorario di un angolo espresso in gradi:

```
\includegraphics[width=%
0.5\columnwidth,angle=45]{Mani}
```



Per mantenere ordine nei file sorgenti, è consigliabile raccogliere tutte le figure in una cartella. Se si hanno molte figure, è conveniente suddividere la cartella in sottocartelle; se per esempio tali sottocartelle si chiamano `grafici` e `foto`, è sufficiente inserire nel preambolo

```
\graphicspath{{grafici/},{foto/}}
```

Per evitare problemi, è opportuno che il percorso dei file inclusi mediante il comando `\graphicspath` non contenga spazi.

Se si prevede di compilare sia con $\text{\LaTeX}$ sia con $\text{\PDFLaTeX}$, è consigliabile non specificare l'estensione dei file grafici caricati dal comando `\includegraphics`; volendo per esempio inserire il file `figura.eps`, è necessario creare il file `figura.pdf` (per esempio con Ghostview o GSview) e poi scrivere nel testo sorgente

```
\includegraphics{figura}
```

Così facendo, quando si compila con $\text{\LaTeX}$ viene caricato il file `figura.eps`, mentre quando si compila con $\text{\PDFLaTeX}$ viene caricato il file `figura.pdf`.

6.3 Figure e tabelle in testo e fuori testo

Normalmente, il comando `\includegraphics` e l'ambiente `tabular` vengono usati rispettivamente all'interno dell'ambiente `figure` e `table` per creare una figura o una tabella mobile, che nel documento finale sarà posizionata da $\text{\LaTeX}$ in accordo con determinate opzioni e regole tipografiche, accompagnata da una didascalia (vedi il paragrafo successivo). Essi possono però essere impiegati diversamente, quando si vuole inserire una figura o una tabella in un punto ben preciso del documento e

non si ha bisogno di una didascalia. Per esempio, il comando `\includegraphics` può essere usato per inserire il sigillo della propria università o il logo della propria azienda nella copertina di una relazione.

Il comando `\includegraphics` (come del resto l'ambiente `tabular`) non inizia un nuovo capoverso; esso produce un'unità tipografica indivisibile che viene trattata da $\text{\LaTeX}$ come se fosse un carattere (una *scatola*, nel gergo di $\text{\LaTeX}$), e può quindi essere usato "in linea" e senza racchiuderlo nell'ambiente `figure`, per inserire una piccola immagine oppure un simbolo in una riga di testo, come nel seguente esempio:

```
\emph{Le Chat noir}
\includegraphics[width=
0.15\columnwidth]{LeChatNoir}
(Il Gatto nero) fu un celebre
locale di Montmartre (Parigi).
```



Le Chat noir (Il Gatto nero) fu un celebre locale di Montmartre (Parigi).

In generale, ci sono due tipi di figure (e tabelle):

Una figura (o tabella) in testo appartiene al flusso del discorso e non può essere posizionata altrove senza che il discorso ne risenta.

Una figura (o tabella) fuori testo non appartiene al flusso del discorso e può essere spostata per esigenze tipografiche.

Solo le figure e le tabelle fuori testo ammettono una didascalia e un numero al quale potersi riferire: chi legge deve poter capire che cosa c'è nella figura o nella tabella (questo è lo scopo della didascalia) e deve saperla trovare (con il numero).

Le figure e le tabelle in testo, invece, non ammettono una didascalia proprio perché la loro funzione è spiegata dal testo che le precede e le segue. Non ci devono essere riferimenti a queste direttamente da altre parti del documento. Devono essere di piccole dimensioni, e devono essere di chiarissima comprensione. Una figura o una tabella complessa *devono* andare fuori testo.

Di solito, le figure e le tabelle in testo sono centrate orizzontalmente rispetto alla giustezza del testo. A tal fine, si usa l'ambiente `center`:

```
La figura seguente
\begin{center}
\includegraphics[width=
0.5\columnwidth]{Rettili}
\end{center}
riproduce
l'incisione su legno
\emph{Tassellazione del
piano con rettili},
di M.~Escher
(l'immagine è
tratta da \url{http://
www.mcescher.com/}).
```

La figura seguente



riproduce l'incisione su legno *Tassellazione del piano con rettili*, di M. Escher (l'immagine è tratta da <http://www.mcescher.com/>).

La tabella seguente

```
\begin{center}
\begin{tabular}{cc}
\toprule


$p$  &  $\neg p$


\midrule
V & F \\
F & V \\
\bottomrule
\end{tabular}
\end{center}
```

riproduce la tavola di verità
della negazione logica.

La tabella seguente

p	$\neg p$
V	F
F	V

riproduce la tavola di verità della
negazione logica.

Quando si inseriscono tabelle o figure in testo, o fuori testo mediante gli ambienti `table` o `figure`, è preferibile usare, in luogo del comando `\textwidth` (che indica la larghezza del corpo del testo), la larghezza della colonna di composizione `\columnwidth` (che per documenti scritti in un'unica colonna coincide con `\textwidth`), poiché, così facendo, se si vuole passare da una composizione a piena pagina a una composizione a più colonne non è necessario modificare il codice.

Se si inseriscono tabelle o figure in un elenco o in una citazione, può essere utile servirsi del comando `\linewidth`, che indica la lunghezza della riga corrente: normalmente essa è uguale a `\columnwidth`, ma in alcuni ambienti, come per l'appunto quelli per gli elenchi e per le citazioni, è più corta [2, p. 88].²

6.4 Gli oggetti mobili

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

La soluzione a questo problema è di rendere mobili (in inglese *floating*, “galleggianti”) le tabelle o le figure che non stanno sulla pagina corrente e stamparle su una pagina successiva, riempiendo quella corrente con il testo successivo (tabelle e figure di questo genere vengono dette “fuori testo”). $\text{\LaTeX}$ offre due ambienti per gli oggetti mobili, uno per le tabelle (`table`) e uno per le figure (`figure`).³

Per ottenere il massimo da questi due ambienti è importante comprendere almeno approssimativamente come $\text{\LaTeX}$ tratta internamente gli oggetti mobili. In caso contrario, essi possono diventare una fonte di frustrazione, perché $\text{\LaTeX}$ non li mette mai dove ci si aspetterebbe.

²È preferibile evitare di servirsi del comando `\linewidth` fuori dagli ambienti per gli elenchi e per le citazioni, in quanto il suo valore è, in generale, imprevedibile al di fuori di quei contesti.

³Se si scrive un documento su più colonne, oltre agli ambienti `table` e `figure`, che si usano per tabelle e figure che si estendono per la larghezza di una singola colonna (`\columnwidth`), si possono usare anche gli ambienti `table*` e `figure*`, per tabelle e figure che si estendono lungo l'intera larghezza della pagina (`\textwidth`).

Tabella 6.7: Caratteri di trasferimento.

Carattere	Permesso di spostare l'oggetto
h	Qui (<i>here</i>), se possibile
t	In cima (<i>top</i>) alla pagina
b	In fondo (<i>bottom</i>) alla pagina
p	In una pagina di soli oggetti mobili (<i>page of floats</i>)
!	Senza considerare molti dei parametri interni di $\text{\LaTeX}$

6.4.1 Gli ambienti `table` e `figure`

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

```
\begin{table}[<posizionamento>]
```

oppure

```
\begin{figure}[<posizionamento>]
```

chiamato `<posizionamento>`. Questo parametro serve a dire a $\text{\LaTeX}$ dove gli oggetti possono essere spostati. Un `<posizionamento>` è costituito da una sequenza di *caratteri di trasferimento* (vedi la tabella 6.7). Una tabella mobile potrebbe per esempio iniziare con la seguente riga:

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

In questo esempio, $\text{\LaTeX}$ cercherà di posizionare la tabella in cima alla pagina (t). Se questo non è possibile, prova a posizionarla in fondo alla pagina (b). Se neppure questo è possibile, $\text{\LaTeX}$ stabilisce se si può posizionare la tabella in una pagina di soli oggetti mobili (p). Se ciò non è possibile, $\text{\LaTeX}$ inizia una nuova pagina, e tratta la tabella come se fosse appena comparsa nel testo. (In generale, se il posizionamento di un oggetto non viene specificato, le classi standard assumono che sia proprio `tbp`.)

$\text{\LaTeX}$ colloca ogni oggetto mobile che incontra in base al posizionamento specificato dall'autore. Se un oggetto non può essere posizionato nella pagina corrente, viene accumulato nella coda delle tabelle o delle figure (queste code sono di tipo `FIFO`, *First In First Out*, "primo arrivato, primo servito"). Quando viene iniziata una nuova pagina, $\text{\LaTeX}$ controlla innanzitutto se è possibile una pagina speciale di soli oggetti mobili con quelli presenti nelle code. Se questo non è possibile, il primo oggetto in ciascuna coda è trattato come se fosse appena comparso nel testo: $\text{\LaTeX}$ prova ancora a collocarlo secondo il suo parametro di posizionamento (tranne per `h`, che non è più possibile). Tutti i nuovi oggetti mobili che compaiono nel testo vengono posti nelle rispettive code. $\text{\LaTeX}$ mantiene strettamente l'ordine di inserimento originale per ogni tipo di oggetto mobile. Ecco perché un oggetto che non può essere posizionato spinge tutti gli oggetti successivi alla fine del documento. Se $\text{\LaTeX}$ non mette gli oggetti mobili dove ci si aspettava, è talvolta a causa di un solo oggetto che blocca una delle due code.

Consideriamo un altro esempio:

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

In questo caso, $\text{\LaTeX}$ cercherà *disperatamente* (!) di mettere la figura nel punto esatto in cui è situato il relativo ambiente (h). Se questo non è possibile, prova a posizionare la figura in fondo alla pagina (b). Se neppure questo è possibile, stabilisce se è possibile posizionare la figura in una pagina di soli oggetti mobili (p). Se non c'è abbastanza materiale per una pagina di questo tipo, $\text{\LaTeX}$ inizia una nuova pagina e tratta la figura come se fosse appena comparsa nel testo.

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

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

è possibile specificare una didascalia per l'oggetto mobile. $\text{\LaTeX}$ aggiunge l'intestazione "Tabella" o "Figura" e un numero progressivo. I due comandi

```
\listoftables
```

e

```
\listoffigures
```

agiscono analogamente al comando `\tableofcontents`, stampando l'elenco delle tabelle e delle figure, rispettivamente. In questi elenchi, verrà ripetuta tutta la didascalia. Se si ha la tendenza a usare didascalie lunghe, è consigliabile fornirne una versione più corta, che comparirà negli elenchi. Questo si ottiene specificando la versione breve tra parentesi quadre dopo il comando `\caption`.

```
\caption[<didascalia corta>]{<didascalia>}
```

Mediante i comandi `\label`, `\ref` e `\vref` (vedi il paragrafo 5.9 a pagina 49), si possono creare riferimenti a oggetti mobili nel documento. Il comando `\label`, che deve comparire *dopo* il corrispondente `\caption`, permette di fare riferimento alla tabella tramite `\ref` o `\vref`.

Il modo migliore per introdurre un oggetto mobile è scrivere il relativo ambiente tra capoversi, cioè preceduto e seguito da una riga vuota. Ecco un esempio tipico d'uso dell'ambiente `table`:

```
\dots Qui finisce un capoverso.
```

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

La tabella~\vref{tab:esempio} è un esempio di tabella mobile.

Per centrare una figura mobile è opportuno usare il comando `\centering` invece dell'ambiente `center`, poiché lo spazio verticale supplementare inserito da quest'ultimo — che è adeguato nell'inserimento di figure (e tabelle) in testo — risulta invece indesiderato nell'inserimento di figure mobili (di regola, la didascalia di una figura si scrive *sotto* la figura). Ecco un esempio tipico d'uso dell'ambiente `figure`:

`\dots` Qui finisce un capoverso.

```
\begin{figure}[tb]
\centering
\includegraphics[<...>]{<...>}
\caption{<...>}
\label{fig:esempio}
\end{figure}
```

La figura~\vref{fig:esempio} è un esempio di figura mobile.

Il controllo degli oggetti mobili

Spesso gli utenti ritengono che $\text{\LaTeX}$ sposti le figure (e in generale gli oggetti mobili) lontano dal punto in cui vengono inserite. Nella maggioranza dei casi questo è dovuto a un uso erraneo delle opzioni di posizionamento degli oggetti mobili. In questo testo si vuole evidenziare che alcune scelte devono essere prese nella fase di stesura del testo (paragrafo 6.4.2) mentre altre sono riservate, quando necessarie, alla fase di revisione (paragrafo 6.4.3 nella pagina seguente).

6.4.2 Che cosa fare durante la stesura del testo

In primo luogo bisogna accettare il fatto che se $\text{\LaTeX}$ sposta un oggetto è perché lo spazio è fisicamente insufficiente, o per considerazioni estetiche. Per esempio, $\text{\LaTeX}$ non metterà mai una figura seguita da un titolo di paragrafo e da un cambio pagina, ma preferirà stampare il titolo del paragrafo e poi la figura, oppure, se aggiungiamo un oggetto mobile in fondo a una pagina, $\text{\LaTeX}$ è obbligato a spostarlo almeno nella pagina successiva. Se lo spazio è insufficiente, è inutile cercare di forzare $\text{\LaTeX}$ a mettere l'oggetto mobile in tale posizione: se lo spazio fisico non c'è, non possiamo inventarlo [16, p. 15].

Per fortuna, con un minimo di accortezza $\text{\LaTeX}$ fa un ottimo lavoro. Per prima cosa è opportuno evitare i posizionamenti fatti “a mano”: in fase di redazione bisogna solo concentrarsi sui contenuti e non sull'impaginazione. In generale i posizionamenti fatti a mano interferiscono con il complesso algoritmo di $\text{\LaTeX}$ per il posizionamento degli oggetti mobili e portano a risultati peggiori rispetto a quelli automatici. Seguendo le semplici indicazioni che seguono, gli oggetti mobili vengono mantenuti vicini al punto di inserimento e inoltre si evita che l'utente si preoccupi continuamente del posizionamento degli oggetti mobili, lasciando più tempo per lavorare sui contenuti.

Una delle origini dei problemi lamentati è l'uso eccessivo dell'opzione `h`: gli oggetti mobili vengono spesso inseriti con l'opzione `htbp` o peggio `!ht`. In genere si pensa che questa opzione sia la migliore per mantenere gli oggetti mobili vicino al punto di inserzione. In realtà può funzionare bene solo quando gli oggetti inseriti sono molto piccoli (dove per piccolo si intende con un'altezza molto inferiore rispetto all'altezza del corpo del testo).

Il modo migliore per usare le opzioni di posizionamento è quello di domandarsi in primo luogo se l'oggetto mobile sarà abbastanza piccolo per stare in una pagina di testo o se avrà bisogno di una pagina tutta per sé. Nel primo caso lo introdurremo

quindi con un’opzione di posizionamento `tb`, nel secondo con `p`. Se non ci sono oggetti mobili in sospeso, nel primo caso $\text{\LaTeX}$ potrà spostare l’oggetto subito prima del punto di inserzione, cosa che non può fare se si usa `h` prima di `t`, o nella pagina immediatamente successiva. Usando invece `p` per i grossi oggetti mobili, questi verranno immediatamente stampati in una pagina dedicata, e non verranno spostati alla fine del capitolo come succede con `tbp`. Basta sfogliare un qualunque testo ben impaginato per accorgersi che le tabelle e le figure sono introdotte proprio in questo modo: di regola all’inizio o alla fine della pagina, in una pagina intera se sono grandi, raramente nel corpo del testo se sono davvero piccole.⁴

Infine è utile ricordare che $\text{\LaTeX}$ riesce a posizionare tutti gli oggetti in modo corretto solo se il rapporto testo/oggetti è sufficientemente alto. Da questo segue che è auspicabile (peraltro non solo per fini tipografici) scrivere contenuti interessanti piuttosto che tentare di colmare le lacune con immagini o tabelle. Se tutto ciò non fosse sufficiente, nella fase precedente la stampa, e solamente allora, può essere necessario intervenire manualmente come spiegato nel paragrafo seguente.

6.4.3 Che cosa fare durante la revisione del testo

Nella fase che precede la stampa, può essere necessario intervenire manualmente per correggere il posizionamento degli oggetti mobili. In alcune circostanze potrebbe essere necessario usare i comandi

```
\clearpage
```

oppure

```
\cleardoublepage
```

Essi ordinano a $\text{\LaTeX}$ di stampare immediatamente tutti gli oggetti mobili rimasti nelle code e cominciare una nuova pagina. Il comando `\cleardoublepage` inizia una nuova pagina destra (nel caso di documenti a doppia facciata).

Possono inoltre essere utili il pacchetto `float`, che permette di forzare il posizionamento dell’oggetto nel punto in cui è situato il relativo ambiente per mezzo dell’opzione `H` (“esattamente qui, in ogni caso”), e il pacchetto `placeins`, che permette di mettere delle barriere invalicabili per gli oggetti mobili con il comando `\FloatBarrier`.

⁴Se si scrive un documento su più colonne, i risultati migliori si ottengono scegliendo l’opzione di posizionamento `t` e poi, in fase di revisione, spostando l’inserimento degli oggetti in modo che siano ripartiti su tutte le colonne (altrimenti in alcuni casi $\text{\LaTeX}$ può metterne due o più su una colonna e nessuno su un’altra, producendo un effetto non molto gradevole).

`\dots`
Qui finisce un capoverso.

...Qui finisce un capoverso.

```
\begin{figure}[H]
\centering
\includegraphics[width=
0.5\columnwidth]{Formica}
\caption{Una figura
posizionata manualmente.}
\label{fig:float}
\end{figure}
```



Figura 6.1: Una figura posizionata manualmente.

La figura~\vref{fig:float} è un esempio di figura mobile posizionata manualmente.

La figura 6.1 è un esempio di figura mobile posizionata manualmente.

È opportuno non abusare dell'opzione H fornita dal pacchetto float: essa va usata *solo in circostanze eccezionali ed esclusivamente in sede di revisione finale*, per ottenere un effetto di impaginazione particolare, se si sa davvero che cosa si sta facendo.

6.4.4 Personalizzare le didascalie

Il formato delle didascalie può essere convenientemente controllato con il pacchetto caption, che fornisce molti modi per personalizzare le didascalie negli ambienti mobili come figure e table.

Per esempio, l'opzione font=small consente di avere le didascalie in corpo più piccolo rispetto al testo. L'opzione format=hang rientra il testo delle didascalie formate da almeno due righe, così esso sarà incolonnato (*hang*) sotto la prima riga del testo (le didascalie che occupano una singola riga sono centrate automaticamente da L^AT_EX). Le didascalie di questo documento sono state composte proprio in questo modo:

```
\usepackage[font=small,format=hang]{caption}
```

6.4.5 Affiancare figure o tabelle

Per affiancare più figure o tabelle si usa il pacchetto subfig (il successore di subfigure, scritto dallo stesso autore). Il pacchetto permette di dare a ciascuna sottofigura o sottotabella una didascalia. Richiede la presenza del pacchetto caption.

Il comando \subfloat inserisce una sottofigura o sottotabella; i suoi due argomenti opzionali hanno lo stesso ruolo dell'argomento opzionale e obbligatorio del comando \caption: il primo, se presente, indica ciò che va nell'elenco relativo (\listoffigures o \listoftables), mentre il secondo indica la sottodidascalia da porre nell'ambiente mobile [11]. Ciascun sottooggetto può contenere un comando \label per potersi riferire in seguito a esso.

Per esempio, la figura 6.2 nella pagina seguente è stata ottenuta con un codice del tipo seguente:

```
\begin{figure}[tb]
\centering
```

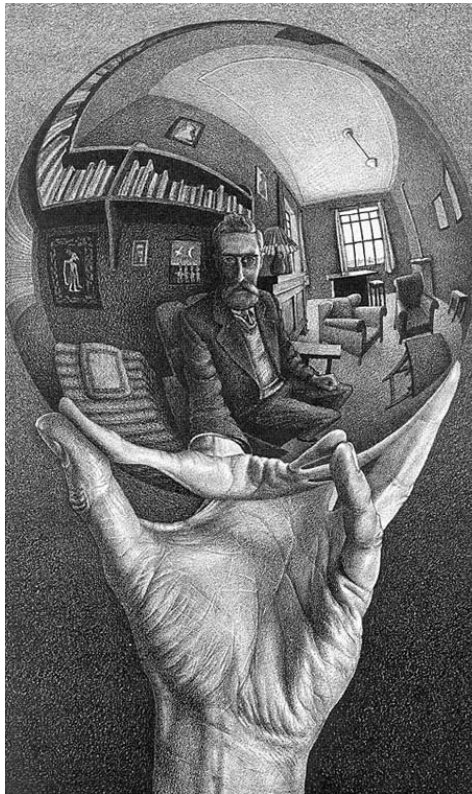
(a) *Mano con sfera riflettente.*(b) *Belvedere.*

Figura 6.2: Un esempio d'uso del pacchetto subfig (le immagini, che riproducono due litografie di M. Escher, sono tratte da <http://www.mcescher.com/>).

```
\subfloat[][\emph{Mano con sfera riflettente}.]
{\includegraphics[width=.45\columnwidth]{Sfera}} \quad
\subfloat[][\emph{Belvedere}.]
{\includegraphics[width=.45\columnwidth]{Belvedere}}
\caption{Due litografie di M.~Escher.}
\label{fig:subfig}
\end{figure}
```

Capitolo 7

La matematica

In questo capitolo verrà esplorato uno dei principali punti di forza di $\text{\LaTeX}$: la composizione di formule matematiche. Questo capitolo intacca solamente la superficie dell'argomento. Benché gli strumenti che verranno spiegati siano sufficienti per la maggior parte delle esigenze, se non si trova la soluzione per “quella” particolare necessità di scrittura matematica è molto probabile che il problema sia risolvibile con una funzione del pacchetto `amsmath` o con qualche altro pacchetto.

La scrittura della matematica è regolata da norme che dipendono prima di tutto dalle tradizioni e dalla cerchia dei lettori cui lo scritto è destinato. Queste regole sono esplicitate nel mondo dalla norme emesse dall'ISO, in Italia dalle norme definite dall'UNI. Le norme UNI hanno valore di legge: se un documento legale, un capitolato d'appalto, ... viene scritto in modo conforme alle norme, mantiene il valore legale che deve avere. Per maggiori dettagli sulle norme ISO-UNI si rimanda alla relativa documentazione e a [2].

7.1 Formule in corpo e fuori corpo

Da qui in avanti, per tutto il capitolo, diamo per scontato che siano caricati i pacchetti `amsmath` e `amssymb`:

```
\usepackage{amsmath,amssymb}
```

Con $\text{\LaTeX}$ ci sono due modi per scrivere la matematica:

Una formula in corpo (o “in linea”) è un'espressione matematica composta da $\text{\LaTeX}$ in linea con il corpo del testo (“incorporata nel testo”), come per esempio

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

Una formula fuori corpo (o “in display”) è un'espressione composta da $\text{\LaTeX}$ in linee a sé stanti, staccate dal testo precedente e seguente mediante spazi di ampiezza adeguata per “mettere in mostra” l'espressione; per esempio

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

Quando una formula è in corpo, $\text{\LaTeX}$ fa il possibile per schiacciarla e non aumentare l'interlinea. Se la stessa formula è fuori corpo c'è molta più libertà di

manovra. È preferibile servirsi delle formule in corpo solo per espressioni di piccole dimensioni: le altre formule vanno messe fuori corpo.

Le formule in corpo si inseriscono includendole tra dollari:

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

Il font delle formule è il corsivo matematico (diverso dal corsivo del testo). Gli esponenti sono stampati più piccoli delle basi.

Le formule fuori corpo si scrivono usando diversi ambienti, di cui i più semplici sono `equation` per le formule numerate e `equation*`, di solito abbreviato in `[...]`, per le formule non numerate.¹

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

$$F'(x) = f(x). \quad (7.1)$$

Si può avere un riferimento a una formula tramite `\label` e `\eqref`:

`\begin{equation}`
`\label{eq:euler}`
 $e^{i\pi} + 1 = 0.$
`\end{equation}`
Dalla formula~`\eqref{eq:euler}`
si deduce che~`\dots`

$$e^{i\pi} + 1 = 0. \quad (7.2)$$

Dalla formula (7.2) si deduce che...

Ci sono delle differenze tra il modo con cui si inserisce la matematica (*modo matematico*) e il modo con cui si inserisce il testo (*modo testuale*). Per esempio, nel modo matematico:

- Gli spazi e le interruzioni di riga non hanno significato, poiché gli spazi sono inseriti automaticamente da $\text{\LaTeX}$, sulla base della struttura dell'espressione matematica: quindi $x+y+z=n$ dà lo stesso risultato di $x + y + z = n$. Se occorre, gli spazi devono essere specificati usando comandi speciali come `\,` (che produce uno spazio sottile) oppure `\quad` o `\qquad` (che danno spazi ampi, detti rispettivamente di un “quadrato” e di un “quadrato”; vedi il paragrafo 7.2.10 a pagina 84).
- Non sono ammesse linee vuote.

¹Nell'inserimento delle formule fuori corpo, l'uso del codice `$$...$$` è invece sempre sconsigliabile per molte ragioni [6, 22].

- Ogni lettera è considerata come un nome di variabile e viene scritta come tale. Se si vuole inserire un breve testo all'interno di una formula fuori corpo (con font e spaziatura normali), allora bisogna scriverlo usando il comando `\text{<...>}`. La spaziatura prima e dopo `\text` va esplicitata.

$$\begin{array}{ll}
 \text{\$x+y+z=n\$} \quad \backslash \quad & x+y+z=n \\
 \text{\$ x + y + z = n \$} & x+y+z=n \\
 \\
 \backslash [& \\
 z^2+1=0 \quad \backslash quad & z^2+1=0 \quad \text{per } z=\pm i. \\
 \backslash \text{per } \$z=\pm i\$ \} & \\
 \backslash] &
 \end{array}$$

Il comando `\pm` dà $\pm$. C'è anche `\mp` che dà $\mp$.

7.2 Nozioni basilari

In questo paragrafo verranno descritti alcuni comandi che vengono usati di frequente per scrivere formule matematiche. Nel paragrafo 7.10 a pagina 98 è riportato un elenco dettagliato dei principali comandi per scrivere simboli matematici.

7.2.1 Raggruppamenti

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

$$\begin{array}{ll}
 \backslash [& \\
 a^{x+y} \quad \backslash neq \quad a^{\{x+y\}} & a^x + y \neq a^{x+y} \\
 \backslash] &
 \end{array}$$

7.2.2 Apici, pedici e radici

Gli apici e i pedici possono essere scritti usando i caratteri `^` e `_`:

$$\begin{array}{ll}
 \text{\$a_1\$} \quad \backslash quad \quad \text{\$x^2\$} \quad \backslash quad & \\
 \text{\$e^{\-\alpha t}\$} \quad \backslash quad & a_1 \quad x^2 \quad e^{-at} \quad a_{ij}^3 \\
 \text{\$a^{\{3\}}_{\{i j\}}\$} \quad \backslash \quad & e^{x^2} \neq e^{x^2} \\
 \text{\$e^{\{x^2\}} \quad \backslash neq \quad \{e^x\}^2\$} &
 \end{array}$$

Non si può scrivere `x_n_k`: è necessario usare le graffe.

Dalla successione `\$x_n\$` estrarre `\$x_{n_k}\$`.
Dalla successione x_n estrarre x_{n_k} .

La radice quadrata si scrive come `\sqrt{x}`, la radice n -esima si ottiene invece con `\sqrt[n]{<n>}{<...>}`. La dimensione della radice è determinata automaticamente da $\text{\LaTeX}$.

$$\begin{array}{ll}
 \text{\$\sqrt{x}\$} \quad \backslash quad & \sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2} \\
 \text{\$\sqrt{x^2 + \sqrt{y}}\$} \quad \backslash quad & \\
 \text{\$\sqrt[3]{2}\$} &
 \end{array}$$

7.2.3 Somme, prodotti e frazioni

Il simbolo di sommatoria è generato con `\sum` e il simbolo di produttoria con `\prod`. Gli estremi si scrivono come pedici e apici.

Trovare il massimo valore
della funzione

```
\[
f(x_1,\dots,x_n)=
\prod_{k=1}^n x_k
\]
```

sotto la condizione

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

Trovare il massimo valore della funzione

$$f(x_1, \dots, x_n) = \prod_{k=1}^n x_k$$

sotto la condizione

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

Una frazione si ottiene con `\frac{\langle numeratore \rangle}{\langle denominatore \rangle}`. A volte la forma n/m è preferibile, perché è più gradevole a vedersi per piccole quantità di “materiale frazionario”.

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

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

7.2.4 Limiti, derivate e integrali

I limiti si fanno con il comando `\lim_{\langle variabile \rangle \to \langle valore \rangle}`.

```
\[
\lim_{x \to 0}
\frac{\sin x}{x} = 1 \quad \quad
\lim_{n \to +\infty} f_n = \delta
\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \quad \lim_{n \rightarrow +\infty} f_n = \delta$$

Il comando `\infty` dà ∞ .

Le derivate si fanno con il simbolo `'` che produce un segno di “primo”.

```
\[
y=x^2, \quad y'=2x, \quad \quad
y''=2.
\]
```

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

Il simbolo di integrale è generato tramite `\int`. Gli estremi di integrazione sono come pedici e apici: `\int_a^b`. Se un pedice o apice è formato da più di una lettera o cifra, va racchiuso tra graffe: `\int_{a+1}^{b+1}`.

```
\[
\int_a^{a+T} f(x) \, dx =
\int_0^T f(x) \, dx
\]
```

$$\int_a^{a+T} f(x) dx = \int_0^T f(x) dx$$

Lo spazio sottile $\int$, allontana dx da $f(x)$.

Per gli integrali multipli si usano i comandi `\iint`, `\iiint`, `\iiiiiint` e `\idotsint`.

```
\[
\iint_D f(x,y)\,dx\,dy \quad \iint_D f(x,y)dx dy \quad \iiint g\,dx\,dy\,dz
\iiint g\,dx\,dy\,dz
\]
```

7.2.5 Insiemi numerici

I simboli degli insiemi numerici si ottengono con `\mathbb` (*blackboard bold*, “neretto da lavagna”).

```
\[
x^2 \geq 0 \quad \forall x \in \mathbb{R}.
\]
```

Il comando `\in` dà la relazione di appartenenza $\in$ (per la non appartenenza $\notin$ si usa `\notin`).

Conviene scrivere nel preambolo le seguenti definizioni:

```
\newcommand{\numberset}{\mathbb}
\newcommand{\N}{\numberset{N}}
\newcommand{\R}{\numberset{R}}
```

In questo modo, basta scrivere `\N` per avere $\mathbb{N}$ ed è possibile cambiare notazione con un’unica modifica.

7.2.6 Lettere greche

Le lettere greche minuscole si ottengono con `\alpha`, `\beta`, `\gamma`, `\delta`, ... (vedi la tabella 7.4 a pagina 98).

Alcune lettere greche minuscole hanno delle forme varianti i cui nomi cominciano con l’abbreviazione *var* (vedi la tabella 7.1 nella pagina seguente); esse negli Stati Uniti sono delle varianti, ma in Europa (a eccezione di `\varsigma`) sono considerate le forme principali. Per scrivere documenti in accordo con il gusto europeo, è utile ridefinire le lettere greche varianti (a eccezione di `\varsigma`) come normali. A tal fine, è sufficiente scrivere nel preambolo

```
\renewcommand{\epsilon}{\varepsilon}
\renewcommand{\theta}{\vartheta}
\renewcommand{\rho}{\varrho}
\renewcommand{\phi}{\varphi}
```

Le lettere greche maiuscole si ottengono con `\Gamma`, `\Delta`, ... (vedi la tabella 7.5 a pagina 98). Non ci sono α e β maiuscole in $\text{\LaTeX} 2_{\epsilon}$, perché sono uguali alle normali A e B latine.

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

Tabella 7.1: Lettere greche minuscole: forme principali e varianti.

ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>
ρ	<code>\rho</code>	ϱ	<code>\varrho</code>
σ	<code>\sigma</code>	ς	<code>\varsigma</code>
ϕ	<code>\phi</code>	φ	<code>\varphi</code>

7.2.7 Accostare simboli ad altri simboli

Per posizionare un simbolo sopra o sotto un altro simbolo si usano i comandi `\overset` e `\underset`.

Il comando `\overset{⟨primo argomento⟩}{⟨secondo argomento⟩}` pone il simbolo indicato nel *⟨primo argomento⟩* sopra quello specificato nel *⟨secondo argomento⟩*, scrivendo il primo con dimensioni inferiori (pari a quelle di un apice), mentre il secondo viene mantenuto nella sua posizione usuale (tipicamente, il *⟨secondo argomento⟩* è un simbolo di relazione binaria). Il comando `\underset` ha un comportamento analogo.

Il simbolo

`\[`
`\overset{H}{=}`
`\]`

indica l'uguaglianza nel senso
del teorema di
de l'Hôpital.

Il simbolo

$\underline{\underline{H}}$

indica l'uguaglianza nel senso del teorema di de l'Hôpital.

7.2.8 Barre e accenti

Il comando `\bar` pone un trattino sul carattere seguente: il simbolo $\bar{x}$, per esempio, indica un nome di variabile distinto da x .

I comandi `\overline` e `\underline` rispettivamente sopralineano e sottolineano tutto quanto si trova tra le graffe: il simbolo $\overline{x}$ indica un operatore (coniugio di numeri complessi, per esempio) applicato alla variabile x .

`\bar{x}` $\quad$ `\bar{X}`
`\overline{m+n}`

$\bar{x}$ $\bar{X}$ $\overline{m+n}$

I comandi `\vec` e `\overrightarrow` sono come `\bar` e `\overline` con frecce al posto di righe.

`\vec{x}` $\quad$
`\overrightarrow{AB}`

$\vec{x}$ $\overrightarrow{AB}$

Ci sono tre tipi di barre verticali, che differiscono per come sono spaziati i simboli che hanno intorno:

- semplice `|` (o `\vert`);
- delimitatore sinistro `\lvert` e destro `\rvert`;

- relazione binaria `\mid` (per la divisibilità e per il “tale che” negli insiemi).

$$F(x) \mid_{\{x=\gamma(t)\}} \quad \quad F(x)|_{x=\gamma(t)} \quad |x|$$

$$\text{Se } p \mid n^2, \text{ allora } p \mid n.$$

Analogamente, per le doppie barre verticali abbiamo: `\|` (o `\Vert`), `\lVert`, `\rVert` e `\parallel`.

La differenza tra due insiemi si fa con il comando `\setminus`.

$$A \setminus B \quad \quad A \setminus B$$

Per aggiungere alle variabili un accento matematico, come un cappello o un segno di tilde, si possono usare i comandi della tabella 7.6 a pagina 98. I segni di cappello o tilde coprenti diversi caratteri sono prodotti tramite `\widehat` e `\widetilde`.

7.2.9 Punti e frecce

Ci sono due tipi di “due punti”, spaziati diversamente:

- semplice `:`, che è spaziato come un’operazione binaria (divisione);
- `\colon`, che è spaziato come un’interpunzione.

$$f \colon \mathbb{R} \rightarrow \mathbb{R} \quad \quad f: \mathbb{R} \rightarrow \mathbb{R}$$

Per inserire punti ellittici in una formula si usa il comando `\dots`, che, automaticamente a seconda del contesto, inserisce i punti sulla linea di base del testo o li centra rispetto alla riga:

$$\begin{aligned} & \left[\begin{array}{l} x_1, \dots, x_n \\ x_1 + \dots + x_n \end{array} \right] \quad \quad \begin{array}{l} x_1, \dots, x_n \\ x_1 + \dots + x_n \end{array} \end{aligned}$$

Si possono trovare altri esempi nel paragrafo 7.5 nella pagina 89.

Oltre alla freccia semplice $\rightarrow$, che si ottiene con il comando `\to`, c’è anche quella col trattino $\mapsto$, che si ottiene con `\mapsto`.

$$\begin{aligned} & f \colon \mathbb{R} \rightarrow \mathbb{R}, \quad \quad f: \mathbb{R} \rightarrow \mathbb{R}, \\ & x \mapsto x^2 \quad \quad x \mapsto x^2 \end{aligned}$$

I comandi `\xleftarrow` e `\xrightarrow` stampano frecce che si estendono automaticamente per accordarsi con grandezze inusuali di apici e pedici. Questi comandi prendono un argomento facoltativo (il pedice) e un argomento obbligatorio (l’apice, che può anche essere vuoto):

$$\begin{aligned} & \left[\begin{array}{l} \xleftarrow{n+\mu-1} \\ \xrightarrow[T]{n \pm i-1} \end{array} \right] \quad \quad \begin{array}{l} \xleftarrow{n+\mu-1} \\ \xrightarrow[T]{n \pm i-1} \end{array} \end{aligned}$$

Tabella 7.2: Spazi in modo matematico.

<code>\,</code>	Spazio sottile positivo
<code>\quad</code>	Spazio di un “quadrato”
<code>\qquad</code>	Spazio di un “quadrato”

I simboli logici vanno usati solo se si sta scrivendo un lavoro di logica (mentre per il resto è preferibile scrivere estesamente “se... allora”, “se e solo se”, ...): il comando `\implies` dà $\Rightarrow$ e `\iff` dà $\Leftrightarrow$ (entrambi con spazi adeguati prima e dopo). I comandi `\land`, `\lor` e `\lnot` producono rispettivamente $\wedge$, $\vee$ e $\neg$.

7.2.10 Spazi in modo matematico

Se gli spazi scelti da $\text{\LaTeX}$ all’interno delle formule non sono soddisfacenti (ma è un caso che si verifica di rado), è possibile modificarli manualmente, inserendo appositi comandi. Come osservato fin qui, i comandi `\quad` o `\qquad` danno spazi ampi. La dimensione di un `\quad` (—) corrisponde all’ampiezza del carattere “M” nel font corrente, mentre `\qquad` (——) produce uno spazio pari a due `\quad`. L’ampiezza di un `\quad` è chiamata anche “quadrato”, mentre quella di un `\qquad` è detta “quadrato”. Il comando `\,` produce uno spazio sottile ($\smash{\,}$), pari a $\frac{3}{18}$ di `\quad` (vedi la tabella 7.2).²

Lo spazio sottile prodotto dal comando `\,` è molto utile in alcune formule. Si confronti, per esempio,

$$\begin{array}{l} \backslash[\\ \int_a^b f(x)\,dx, \quad \backslashquad \\ \sqrt{2}\,a. \\ \backslash] \end{array} \qquad \int_a^b f(x)dx, \quad \sqrt{2}a.$$

con

$$\begin{array}{l} \backslash[\\ \int_a^b f(x) \, dx, \quad \backslashquad \\ \sqrt{2} \, a. \\ \backslash] \end{array} \qquad \int_a^b f(x)dx, \quad \sqrt{2}a.$$

Nel secondo caso viene una spaziatura (leggermente) insufficiente.

7.3 Gli operatori

Le funzioni matematiche come $\sin$, $\cos$ e $\log$ sono (per tradizione conservata dalle norme ISO-UNI) stampate in tondo per renderne più immediata la visibilità rispetto alle variabili matematiche, che sono stampate in corsivo matematico.

Inoltre, le funzioni matematiche richiedono certi spazi alla loro destra e alla loro sinistra; questi spazi dipendono dalla natura degli oggetti matematici che precedono o seguono le funzioni.

²Oltre ai comandi citati ne esistono altri, come per esempio `\:` o `\;` (che producono uno spazio orizzontale rispettivamente di $\frac{4}{18}$ e di $\frac{5}{18}$ di `\quad`), che però, per motivi di uniformità tipografica, non dovrebbero *mai* essere usati. Anche l’uso, pure possibile, dei comandi `\quad` e `\qquad` fuori dagli ambienti matematici è fortemente sconsigliato.

$\backslash$	
$\cos 2x =$	$\cos 2x = \frac{1 - \sin^2 x}{2},$
$\frac{1 - \sin^2 x}{2},$	
$\backslash$	
$\log \log x, \quad \log(x+y)$	$\log \log x, \quad \log(x+y)$
$\backslash$	

In $\text{\LaTeX}$, i comandi come $\sin$, $\cos$ e $\log$ sono detti *operatori*. Si noti che nella formula $\cos 2x$, fra $\cos$ e 2 c'è più spazio che fra 2 e x . Inoltre, nella formula $\log \log x$ c'è uno spazio sottile tra i due $\log$ e un altro spazio sottile tra $\log$ e x , mentre nella formula $\log(x+y)$ non ci sono spazi tra $\log$ e “(”.

Se si omette il “\” si ha $\cos x$ in corsivo e spaziato scorrettamente, che non è più un operatore. Solo usando gli appositi comandi, $\text{\LaTeX}$ sa che sta usando degli operatori e sa quale font e quali spazi usare. I seguenti sono alcuni operatori predefiniti:

$\arccos x$, $\exp x$,	$\arccos x, \exp x, \min_{x \in A} f(x), \det A, \log x,$
$\min_{x \in A} f(x)$,	$\tan x$
$\det A$, $\log x$, $\tan x$	

Per quel che riguarda la relazione di congruenza modulo m , ci sono due comandi, $\bmod$ e $\pmod$:

$a \bmod b \quad \quad \quad a \bmod b$	$a \equiv b \pmod{m}$
$a \equiv b \pmod{m}$	

La tabella 7.3 nella pagina seguente riporta l'elenco degli operatori predefiniti di $\text{\LaTeX}$.

Dal momento che nelle pubblicazioni relative alla matematica si introducono continuamente nuove funzioni, è utile disporre di un metodo generale per definire nuovi operatori. A tal fine, si usa il comando $\text{\DeclareMathOperator}$.

Per esempio, per definire la funzione matematica arcsinh che denoti l'arcoseno iperbolico (che non è definito né da $\text{\LaTeX}$ né da amsmath), si scrive nel preambolo $\text{\DeclareMathOperator{\arcsinh}\arcsinh}$

Nel seguito, basta scrivere $\text{\arcsinh } x$ per avere $\text{arcsinh } x$ nel font corretto e adeguatamente spaziato su entrambi i lati.

L'operatore che denota la parte reale di un numero complesso è $\text{\Re}$, che produce il simbolo $\Re$; se lo si vuole ridefinire in modo che esso sia scritto in tondo e non in gotico, si usano i comandi

```
\let\Re\relax
\DeclareMathOperator{\Re}{Re}
```

La dichiarazione $\text{\let\Re\relax}$ è necessaria dal momento che l'operatore $\text{\Re}$ è già definito.

Se il nuovo operatore dovesse esser dotato di pedici e apici posizionati “alla maniera dei limiti”, al di sopra e al di sotto come per $\lim$, $\sup$, o $\max$, si userà la forma $*$ del comando $\text{\DeclareMathOperator}$:

```
\DeclareMathOperator*{\Lim}{Lim}
```

Tabella 7.3: Gli operatori predefiniti.

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

7.4 Le parentesi

Per quel che riguarda le parentesi e gli altri delimitatori, $\text{\LaTeX}$ offre un vasto assortimento di simboli. Le parentesi tonde e quadre possono essere scritte “normalmente”, ossia con i caratteri corrispondenti, mentre le parentesi graffe vanno scritte con `\{`; tutti gli altri delimitatori vengono generati tramite comandi speciali (come per esempio `\updownarrow`).

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

Per l’elenco di tutti i delimitatori disponibili si vedano le tabelle 7.10 e 7.11 a pagina 100.

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

I comandi `\bigl` e `\bigr` ingrandiscono lievemente le parentesi:

```
\[
\bigl( (x-y)+(x+y) \bigr)      ((x-y)+(x+y))
\]
```

I comandi `\Bigl` e `\Bigr` producono parentesi ancora più grandi:

```
\[
\Bigl(1+\frac{1}{n}\Bigr)^n      \left(1+\frac{1}{n}\right)^n
\]
```

I comandi `\biggl` e `\biggr` le producono ancora più grandi:

```
\[
\biggl(\sum_n x_n^2\biggr)^{1/2} \left(\sum_n x_n^2\right)^{1/2}
\]
```

Se non basta, ci sono anche `\Biggl` e `\Biggr`.³

³È possibile specificare automaticamente le dimensioni di un delimitatore mediante i comandi `\left` (davanti a un delimitatore di apertura) e `\right` (davanti al corrispondente delimitatore di chiusura); si noti che occorre chiudere ogni `\left` con un corrispondente `\right`, che la dimensione viene correttamente determinata solo se tutti e due sono sulla stessa riga e che se non si vuole niente sulla destra si usa l’invisibile `\right.` (con punto finale). Tuttavia, la scelta manuale è spesso la migliore, dal momento che in molti casi i comandi `\left` e `\right` inseriscono spaziature non richieste e parentesi più grandi del necessario.

7.5 Vettori e matrici

I vettori si scrivono di solito in neretto (corsivo, secondo le norme norme ISO-UNI) oppure in semplice corsivo matematico; talvolta, soprattutto nei testi di fisica, vengono stampati con una freccia sopra. Per scrivere simboli in neretto si può usare il comando `\mathbf`; per il neretto corsivo c'è il comando `\bm`, messo a disposizione dal pacchetto `bm` (*bold math*); per comporre simboli con una freccia sopra si usa il comando `\vec`. Può convenire ridefinire il comando `\vec`, nel preambolo:

```
\renewcommand{\vec}{\bm}
```

In questo modo, basta scrivere `\vec{v}` per avere $\mathbf{v}$ ed è possibile cambiare notazione con un'unica modifica (vedi anche il paragrafo 7.7 a pagina 92).

Per scrivere matrici si usano gli ambienti `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` e `Vmatrix`, che hanno come delimitatori rispettivamente parentesi tonde, parentesi quadre (*braces*), parentesi graffe (*curly Braces*), barre verticali e doppie barre Verticali; c'è anche l'ambiente `matrix` senza delimitatori.

Gli elementi sono centrati automaticamente. Righe e colonne si scrivono come in `tabular`:

- i vari `&` separano gli elementi di una riga;
- il comando `\\` serve per terminare una riga (eccetto l'ultima);
- gli spazi non espliciti sono ignorati.

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

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

```
\[
\begin{vmatrix}
1-x & 2 \\
3 & 4-x
\end{vmatrix}
```

Il seguente è un esempio di matrice con punti ellittici:

```
\[
A=
\begin{bmatrix}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{bmatrix}
\]
```

$$A = \begin{bmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Il comando `\vdots` fa tre punti ellittici verticali, il comando `\ddots` fa tre punti ellittici diagonali. Il comando `\hdotsfor{<n>}` riempie di punti ellittici per n colonne.

```
\[
\begin{bmatrix}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
a_{2n} & & & \\
\hdotsfor{4} \\
a_{n1} & a_{n2} & \dots & a_{nn}
\end{bmatrix}
\]
```

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Per scrivere una piccola matrice in linea con il corpo del testo è disponibile l'ambiente `smallmatrix`:

```
Sia $A=\bigl(
\begin{smallmatrix}
a & b \\
c & d
\end{smallmatrix}
\bigr)$
una matrice invertibile.
```

Sia $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ una matrice invertibile.

Le parentesi vanno aggiunte manualmente.

7.6 Formule fuori corpo

Le formule lunghe non vengono automaticamente divise in parti da $\text{\LaTeX}$. Solo chi ha scritto la formula, infatti, ne conosce il ritmo di lettura e sa dove è più opportuno andare a capo e se allineare o meno le varie righe.

In generale, tuttavia, vale la regola per cui si può andare a capo dopo i simboli di relazione e, subordinatamente, dopo i simboli di relazione e operazione binaria (vedi le tabelle 7.7 e 7.8 a pagina 99), mai dopo gli operatori funzionali (tabella 7.3 a pagina 86), i grandi operatori e i delimitatori di apertura (vedi le tabelle 7.9, 7.10 e 7.11 a pagina 100).

Per spezzare le formule fuori corpo, il pacchetto `amsmath` mette a disposizione (fra l'altro) gli ambienti `multline`, `split`, `gather` e `align`.⁴

⁴Per spezzare formule fuori corpo, l'uso dell'ambiente `eqnarray` è invece sempre sconsigliabile per molte ragioni [6, 22].

7.6.1 Formule spezzate senza allineamento: l'ambiente `multline`

Per dividere una *singola formula* in più righe, *senza particolari allineamenti* fra le varie righe, si usa l'ambiente `multline`.

<code>\begin{multline}</code>	
<code>f=a+b+c \\</code>	$f = a + b + c$
<code>+d+e+g+h \\</code>	$+ d + e + g + h$
<code>+r+s+t.</code>	$+ r + s + t.$
<code>\end{multline}</code>	(7.3)

In stampa la prima riga è allineata a sinistra, l'ultima è allineata a destra, e le altre vengono centrate (a meno che non sia in funzione l'opzione `fleqn`, che compone le formule visualizzate allineandole a sinistra (rispetto a un margine rientrato) invece che centrandole. Il numero della formula è stampato a margine destro dell'ultima riga.

L'ambiente `multline*` produce una formula non numerata.

7.6.2 Formule spezzate con allineamento: l'ambiente `split`

Per dividere una *singola formula* in più righe, *con le righe da allineare*, si usa l'ambiente `split`.

<code>\begin{equation}</code>	
<code>\begin{split}</code>	
<code>a &= b+c-d \\</code>	$a = b + c - d$
<code>&= e-f \\</code>	$= e - f$
<code>&= g+h \\</code>	$= g + h$
<code>&= i.</code>	$= i.$
<code>\end{split}</code>	(7.4)
<code>\end{equation}</code>	

Il simbolo `&` marca il punto di una riga da incolonnare con i punti corrispondenti delle altre righe. Il numero della formula è centrato verticalmente.

A differenza di altre strutture per comporre formule, l'ambiente `split` non produce numeri, poiché è progettato per essere usato esclusivamente all'interno di qualche altra struttura per formule fuori corpo (di solito un ambiente `equation`, `align` o `gather`), che fornisce la numerazione.

7.6.3 Gruppi di formule senza allineamento: l'ambiente `gather`

L'ambiente `gather` serve per raggruppare *più formule*, ognuna centrata in una riga a sé, numerata separatamente, *senza allineamento* l'una con l'altra; a ciascuna formula può essere attribuita un'etichetta.

<code>\begin{gather}</code>	
<code>a=b+c, \\</code>	$a = b + c,$
<code>V+F-S=2.</code>	$V + F - S = 2.$
<code>\end{gather}</code>	(7.5)
	(7.6)

Con più ambienti `equation` di seguito l'effetto sarebbe simile, ma la spaziatura fra le righe sarebbe esagerata. Una formula che finisce con `\notag` non viene numerata.

L'ambiente `gather*` (come `equation*`) non numera le formule.

7.6.4 Gruppi di formule con allineamento: l'ambiente `align`

L'ambiente `align` permette di disporre *gruppi di due o più formule*, ciascuna su una riga, numerate singolarmente, *da allineare fra loro*.

```
\begin{align}
a_1 &= b_1 + c_1 + d_1, \\
a_2 &= b_2, \notag \\
a_3 &= b_3 + c_3.
\end{align}
```

$$a_1 = b_1 + c_1 + d_1, \quad (7.7)$$

$$a_2 = b_2, \quad (7.8)$$

$$a_3 = b_3 + c_3.$$

Il simbolo `&` marca la posizione di una riga da incolonnare (di solito per gli allineamenti vengono scelti i simboli di relazione). Una formula che finisce con `\notag` non è numerata.

L'ambiente `align*` produce formule non numerate.

L'ambiente `align` può anche allineare più colonne di formule. In tal caso i vari `&` hanno due significati diversi:

- i simboli `&` di posto dispari (primo, terzo, ...) segnano i punti da incolonnare;
- i simboli `&` di posto pari (secondo, quarto, ...) servono solo per distanziare una colonna di formule dall'altra.

```
\begin{align}
a &= b, & c &= d, & e &= f, \\
u &= v, & w &= x, & y &= z.
\end{align}
```

$$a = b, \quad c = d, \quad e = f, \quad (7.9)$$

$$u = v, \quad w = x, \quad y = z. \quad (7.10)$$

7.6.5 Gli ambienti `gathered` e `aligned`

Come `equation`, gli ambienti per disporre gruppi di formule `gather` e `align` sono progettati per produrre strutture aventi lunghezza complessiva pari alla lunghezza di una riga; questo implica, per esempio, che non è facile aggiungere parentesi attorno alle strutture. Vengono allora fornite le varianti `gathered` e `aligned`, la cui lunghezza totale è pari alla reale lunghezza dei contenuti; le strutture così ottenute possono quindi essere usate come componenti di un'espressione più complessa. Per esempio:

```
\[
\left.
\begin{aligned}
\nabla \cdot E &= 4\pi\rho, \\
\nabla \cdot B &= 0,
\end{aligned}
\right\} \text{eq. di Maxwell}
```

$$\left. \begin{aligned} \nabla \cdot E &= 4\pi\rho, \\ \nabla \cdot B &= 0, \end{aligned} \right\} \text{eq. di Maxwell}$$

7.6.6 Casi e numerazione subordinata

L'ambiente `cases` serve per le definizioni fatte per casi. La graffa e l'allineamento sono automatici; il testo nella seconda colonna va dentro a `\text{\langle...\rangle}`.

```
\[
f(n):=
\begin{cases}
2n+1, & \text{se } n \text{ è} \\
\text{dispari,} \\
n/2, & \text{se } n \text{ è pari.}
\end{cases}
\]
```

$$f(n) := \begin{cases} 2n+1, & \text{se } n \text{ è dispari,} \\ n/2, & \text{se } n \text{ è pari.} \end{cases}$$

Per numerare un gruppo di formule con uno schema subordinato è disponibile l'ambiente `subequations`:

```
\begin{subequations}
\label{eqn:schema}
\begin{align}
a &= b+c, \\
c &= d, \\
e &= f+g.
\end{align}
\end{subequations}
```

$$a = b + c, \quad (7.11a)$$

$$c = d, \quad (7.11b)$$

$$e = f + g. \quad (7.11c)$$

Le formule (7.11), e in particolare la (7.11b), ...
 e in particolare la (7.11b), ...

7.7 Modificare lo stile e il corpo dei font

In modo matematico, $\text{\LaTeX}$ seleziona automaticamente lo stile e la dimensione dei font in accordo col contesto. Ciò nonostante, a volte è necessario modificare lo stile e il corpo dei caratteri.

<code>\\$x+y+2^n M\cos t\\$ \\</code>	$x+y+2^n M \cos t$
<code>\\$ \mathit{x+y+2^n M\cos t} \\$ \\</code>	$x+y+2^n M \cos t$
<code>\\$ \mathbf{x+y+2^n M\cos t} \\$ \\</code>	$\mathbf{x+y+2^n M \cos t}$
<code>\\$ \mathrm{x+y+2^n M\cos t} \\$ \\</code>	$x+y+2^n M \cos t$
<code>\\$ \mathtt{x+y+2^n M\cos t} \\$ \\</code>	$x+y+2^n M \cos t$
<code>\\$ \mathsf{x+y+2^n M\cos t} \\$</code>	$x+y+2^n M \cos t$

Come si può constatare, i comandi di cambiamento di stile agiscono sia sulle lettere sia sui numeri.

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

Siano <code>\mathcal{C}</code> la	Siano $\mathcal{C}$ la circonferenza di centro O e
circonferenza di centro O	raggio 1, $\mathcal{D}_1$ e $\mathcal{D}_2$ due rette.
e raggio 1 , <code>\mathcal{D}_1</code> e	
<code>\mathcal{D}_2</code> due rette.	

Non bisogna abusare, come spesso fanno i compositori amatoriali, della possibilità di inserire simboli in neretto, tipograficamente piuttosto “pesante”. Ciò

premessi, il comando `\mathbf` è usato comunemente per ottenere lettere latine in neretto in modo matematico, ma per la maggior parte degli altri tipi di simboli matematici non ha effetto, o i suoi effetti dipendono in maniera non prevedibile dalla serie di font matematici in uso. Per esempio, scrivendo

```
\[
\mu, M \quad \quad \mu, M
\mathbf{\mu}, \mathbf{M}
\]
```

si osserva che il comando `\mathbf` non ha funzionato con la lettera μ . Inoltre, le lettere restituite dal comando `\mathbf` sono in tondo, mentre di solito i simboli matematici sono in corsivo.

Per comporre simboli matematici in neretto corsivo è consigliabile servirsi dell'apposito comando `\bm`, fornito dal pacchetto `bm` (*bold math*). Va tenuto presente che `\bm` funziona se (e solo se) il font matematico in uso in quel momento dispone di una versione in neretto di quel simbolo.

```
\[
\mu, M \quad \quad \mu, M
\bm{\mu}, \bm{M}
\]
```

I pedici e gli apici letterali vanno scritti in corsivo matematico se rappresentano quantità variabili (cioè se sono dei *simboli*), in tondo se rappresentano delle apposizioni di una grandezza fisica (cioè se sono semplice *testo*). In quest'ultimo caso si usa il comando `\textup`.

```
\[
V_{\textup{eff}} \quad \psi_{\textup{incidente}}
\]
```

Si confronti:

```
$V_{\textup{eff}}$ \quad V_{\textup{eff}}
$V_{\textup{eff}}$ \quad V_{eff}
```

Se si scrive `V_{eff}`, il font non è il tondo ma il corsivo matematico e la spaziatura risulta completamente sbagliata, dal momento che $\text{\LaTeX}$ interpreta le tre lettere come variabili da moltiplicare: la scrittura corretta è `V_{\textup{eff}}`.

In alternativa, l'opzione `italian` di `babel` offre i comandi `\ap` e `\ped` per scrivere in tondo apici e pedici, rispettivamente.⁵

In modo matematico la dimensione del font può essere impostata manualmente con le quattro dichiarazioni `\displaystyle`, `\textstyle`, `\scriptstyle` e `\scriptscriptstyle`.

⁵Entrambi i comandi funzionano anche in modo testo per inserire pedici o apici nel font corrente. Servono per abbreviazioni antiche come F^{li} oppure per loghi come $\text{Word}_{\text{Star}}$ oppure per gli ordinali maschili o femminili 1^{o} , 2^{a} , ..., che si ottengono rispettivamente con `F\ap{11i}`, `Word\ped{Star}` e `1\ap{o}`, `2\ap{a}`.

$\sum_{k=1}^n z^k$	$\sum_{k=1}^n z^k$
$\sum_{k=1}^n z^k$	$\sum_{k=1}^n z^k$
$\sum_{k=1}^n z^k$	$\sum_{k=1}^n z^k$
$\sum_{k=1}^n z^k$	$\sum_{k=1}^n z^k$
$\sum_{k=1}^n z^k$	$\sum_{k=1}^n z^k$

Il cambiamento di stile influisce anche sul modo in cui vengono scritti gli apici e i pedici.

Esistono anche i comandi `\dfrac` e `\tfrac` come convenienti abbreviazioni per $\{\displaystyle\frac{\dots}\}$ e $\{\textstyle\frac{\dots}\}$.

$\frac{1}{k} \log_2 c(f)$	$\frac{1}{k} \log_2 c(f)$
$\frac{1}{k} \log_2 c(f)$	$\frac{1}{k} \log_2 c(f)$
$\frac{1}{k} \log_2 c(f)$	$\frac{1}{k} \log_2 c(f)$
$\frac{1}{k} \log_2 c(f)$	$\frac{1}{k} \log_2 c(f)$

7.8 Enunciati e dimostrazioni

In questo paragrafo diamo per scontato che sia caricato il pacchetto `amsthm`.

Componendo documenti matematici, è utile disporre di un metodo per introdurre e numerare definizioni, teoremi e strutture simili. I tipi di enunciati non sono predefiniti, ma vanno dichiarati dall'utente, che deve prendere alcune decisioni "globali":

- quali tipi di enunciati si useranno, per esempio definizioni e teoremi;
- il nome dell'ambiente $\text{\LaTeX}$ di ogni tipo di enunciato, per esempio definizione e teorema (`def` non si può usare perché è un comando di base interno di $\text{\TeX}$);
- quali parole esattamente andranno in stampa come titoli dei vari enunciati, per esempio "Definizione" e "Teorema".

Le dichiarazioni globali si fanno con il comando `\newtheorem` (nel preambolo), che consente due forme di definizione:

`\newtheorem{<nome dell'enunciato>}{<titolo>}[<sezione>]`

oppure, in alternativa,

```
\newtheorem{<nome dell'enunciato>}[<numerato come>]{<titolo>}
```

L'argomento *<nome dell'enunciato>* è una parola chiave usata per identificare successivamente l'enunciato. Con l'argomento *<titolo>* invece viene indicato il titolo dell'enunciato, che poi verrà stampato sul documento. Gli argomenti tra parentesi quadre sono opzionali; entrambi vengono impiegati per specificare la numerazione usata per l'enunciato. L'argomento *<numerato come>* deve specificare il nome di un enunciato precedentemente dichiarato: il nuovo enunciato verrà allora numerato con la stessa sequenza numerica. L'argomento *<sezione>* permette di specificare a quale tipo di unità di sezionamento (di regola *chapter* o *section*) collegare la numerazione dell'enunciato.

Un enunciato comincia con il comando `\begin{<nome dell'enunciato>}` e finisce con `\end{<nome dell'enunciato>}`. Un'eventuale specificazione dell'enunciato va fra quadre subito dopo `\begin{<nome dell'enunciato>}`; in stampa verrà fra tonde.

Ci sono tre stili predefiniti di enunciati (*plain*, *definition* e *remark*). I dettagli tipografici di questi stili dipendono dalla classe del documento, ma tipicamente lo stile *plain* produce un corpo del testo in corsivo, mentre gli altri due producono un corpo del testo in tondo. L'elenco seguente riporta le strutture normalmente associate ai diversi stili di enunciati:

plain si usa per teoremi, lemmi, corollari, proposizioni, congetture, criteri, leggi, algoritmi;

definition è adatto per definizioni, condizioni, problemi, esempi;

remark si usa per osservazioni, note, notazioni.

Lo stile predefinito è *plain*. Per specificare stili differenti, si usa il comando `\theoremstyle`, suddividendo gli enunciati per gruppi e scrivendo prima di ogni gruppo l'appropriato `\theoremstyle`. (Qualora i tre stili predefiniti non fossero sufficienti, il pacchetto *amsthm* fornisce il comando `\newtheoremstyle`, che permette di creare stili personalizzati.)

La teoria dovrebbe essere sufficiente, i seguenti esempi dovrebbero chiarire quanto appena spiegato. Se scriviamo nel preambolo

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

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

gli ambienti *definizione* e *teorema* si usano nel modo seguente:

```

\begin{definizione}[di Lord
Kelvin]
Si dice \emph{matematico} colui
per il quale è ovvio che
 $\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}$ .
\end{definizione}
\begin{teorema}
I matematici, se ce ne sono,
sono molto rari.
\end{teorema}

```

Definizione 1 (di Lord Kelvin). Si dice *matematico* colui per il quale è ovvio che $\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}$.

Teorema 1. *I matematici, se ce ne sono, sono molto rari.*

```

Il seguente teorema è a tutti
ben noto.
\begin{teorema}[di Pitagora]
La somma dei quadrati costruiti
sui cateti è uguale
al quadrato costruito
sull'ipotenusa.
\end{teorema}
La dimostrazione è lasciata per
esercizio.

```

Il seguente teorema è a tutti ben noto.

Teorema 2 (di Pitagora). *La somma dei quadrati costruiti sui cateti è uguale al quadrato costruito sull'ipotenusa.*

La dimostrazione è lasciata per esercizio.

Come si vede, $\text{\LaTeX}$ numera automaticamente ogni enunciato e lo stacca da ciò che precede e da ciò che segue. Ogni tipo di enunciato è numerato a parte e non c'è alcun rientro prima del titolo dell'enunciato. Il titolo e il numero dell'enunciato sono in neretto (con punto finale). Il corpo dell'enunciato è in tondo per le definizioni e in corsivo per i teoremi.

L'ambiente `proof` serve per le dimostrazioni.

```

\begin{teorema}[Sorpresa]
Si ha che  $0=1$ .
\end{teorema}
\begin{proof}
Da  $e^{2n\pi i}=1$  segue che
 $e^{1+2n\pi i}=e$ , poi
 $(e^{1+2n\pi i})^{1+2n\pi i}=e^{1+2n\pi i}=e$ , per cui
 $e^{1+4n\pi i-4n^2\pi^2}=e$ , e
infine  $e^{4n\pi i-4n^2\pi^2}=1$ . Mandando
 $n \rightarrow +\infty$  si ha la tesi.
\end{proof}

```

Teorema 3 (Sorpresa). *Si ha che $0=1$.*

Dimostrazione. Da $e^{2n\pi i}=1$ segue che $e^{1+2n\pi i}=e$, poi $(e^{1+2n\pi i})^{1+2n\pi i}=e^{1+2n\pi i}=e$, per cui $e^{1+4n\pi i-4n^2\pi^2}=e$, e infine $e^{4n\pi i-4n^2\pi^2}=1$. Mandando $n \rightarrow +\infty$ si ha la tesi. $\square$

Viene un quadratino a fine dimostrazione.

Vediamo ora un esempio di due enunciati numerati con la stessa sequenza numerica. Se scriviamo nel preambolo

```

\theoremstyle{plain}
\newtheorem{legge}{Legge}
\newtheorem{decreto}[legge]{Decreto}

```

gli ambienti `legge` e `decreto` si usano come segue:

```

\begin{legge}
\label{lex:capo}
Il capo ha ragione.
\end{legge}
\begin{decreto}[Aggiornamento
alla legge~\ref{lex:capo}]
Il capo ha \emph{sempre} ragione.
\end{decreto}
\begin{legge}
Se il capo ha torto, vedere la
legge~\ref{lex:capo}.
\end{legge}

```

Legge 1. *Il capo ha ragione.*

Decreto 2 (Aggiornamento alla legge 1).
Il capo ha sempre ragione.

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

L’enunciato “Decreto” usa lo stesso contatore dell’enunciato “Legge”, perciò ha un numero di identificazione che segue la stessa sequenza numerica usata da questo. Gli enunciati possono essere identificati con un’etichetta.

Se si desidera introdurre un enunciato “Murphy” la cui numerazione sia collegata al paragrafo corrente, è sufficiente scrivere nel preambolo

```
\newtheorem{murphy}{Murphy}[section]
```

L’ambiente `murphy` così definito si usa nel solito modo:

```

\begin{murphy}
Se esistono due o più modi
per fare una cosa, e se uno
di questi modi può creare
una catastrofe, allora
qualcuno lo sceglierà.
\end{murphy}

```

Murphy 7.8.1. *Se esistono due o più modi per fare una cosa, e se uno di questi modi può creare una catastrofe, allora qualcuno lo sceglierà.*

7.9 Diagrammi commutativi

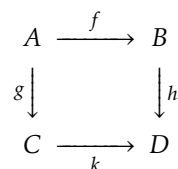
I diagrammi commutativi sono particolari oggetti a metà strada fra un insieme di espressioni matematiche e un disegno che le mette in relazione le une con le altre.

Il pacchetto `amscd` definisce un ambiente `CD` con cui è possibile disegnare diagrammi commutativi piani e senza frecce diagonali. Per diagrammi più complessi il lettore deve rivolgersi a pacchetti più sofisticati, come `PSTricks` oppure `Xy-pic`.

```

\[
\begin{CD}
A @>f>> B \\
@V{g}VV @VV{h}V \\
C @>>k> D
\end{CD}
\]

```



L’ambiente `CD` va usato solo all’interno di un ambiente matematico per formule fuori corpo, come `equation` oppure uno degli altri appositi ambienti visti in questo capitolo.

7.10 Elenco dei simboli matematici

Nelle seguenti tabelle si possono trovare tutti i simboli ai quali si può accedere normalmente dall'ambiente matematico. Queste tabelle sono prese da [18, p. 52-58]. (Per un elenco esauriente dei simboli di $\text{\LaTeX}$, vedi [19].)

Tabella 7.4: Lettere greche minuscole.

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

Tabella 7.5: Lettere greche maiuscole.

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

Tabella 7.6: Accenti in modo matematico.

$\bar{a}$	<code>\bar{a}</code>	$\hat{a}$	<code>\hat{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>
$\vec{a}$	<code>\vec{a}</code>	$\dot{a}$	<code>\dot{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>
$\acute{a}$	<code>\acute{a}</code>	$\grave{a}$	<code>\grave{a}</code>	$\breve{a}$	<code>\breve{a}</code>
$\check{a}$	<code>\check{a}</code>	$\widehat{A}$	<code>\widehat{A}</code>	$\widetilde{A}$	<code>\widetilde{A}</code>

Tabella 7.7: Relazioni binarie.

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

Tabella 7.8: Operazioni binarie.

$+$	<code>+</code>	$-$	<code>-</code>	$\triangleleft$	<code>\triangleleft</code>
$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$\triangleright$	<code>\triangleright</code>
$\cdot$	<code>\cdot</code>	$\div$	<code>\div</code>	$\star$	<code>\star</code>
$\times$	<code>\times</code>	$\setminus$	<code>\setminus</code>	$\ast$	<code>\ast</code>
$\cup$	<code>\cup</code>	$\cap$	<code>\cap</code>	$\circ$	<code>\circ</code>
$\sqcup$	<code>\sqcup</code>	$\sqcap$	<code>\sqcap</code>	$\bullet$	<code>\bullet</code>
$\vee$	<code>\vee o \lor</code>	$\wedge$	<code>\wedge o \land</code>	$\diamond$	<code>\diamond</code>
$\oplus$	<code>\oplus</code>	$\ominus$	<code>\ominus</code>	$\uplus$	<code>\uplus</code>
$\odot$	<code>\odot</code>	$\oslash$	<code>\oslash</code>	$\amalg$	<code>\amalg</code>
$\otimes$	<code>\otimes</code>	$\bigcirc$	<code>\bigcirc</code>	$\dagger$	<code>\dagger</code>
$\triangleup$	<code>\bigtriangleup</code>	$\triangledown$	<code>\bigtriangledown</code>	$\ddagger$	<code>\ddagger</code>
$\triangleleft$	<code>\lhd</code>	$\triangleright$	<code>\rhd</code>	$\wr$	<code>\wr</code>
$\trianglelefteq$	<code>\unlhd</code>	$\trianglerighteq$	<code>\unrhd</code>		

Tabella 7.9: Grandi operatori.

Σ	<code>\sum</code>	$\prod$	<code>\prod</code>	$\coprod$	<code>\coprod</code>
$\int$	<code>\int</code>	$\oint$	<code>\oint</code>	$\bigcup$	<code>\bigcup</code>
$\bigcap$	<code>\bigcap</code>	$\bigvee$	<code>\bigvee</code>	$\bigwedge$	<code>\bigwedge</code>
$\bigoplus$	<code>\bigoplus</code>	$\bigotimes$	<code>\bigotimes</code>		
	<code>\biguplus</code>	$\bigodot$	<code>\bigodot</code>		

Tabella 7.10: Delimitatori.

$($	<code>(</code>	$[$	<code>[o \lbrack</code>	$\{$	<code>\{ o \lbrace</code>
$)$	<code>)</code>	$]$	<code>] o \rbrack</code>	$\}$	<code>\} o \rbrace</code>
$\langle$	<code>\langle</code>	$\lfloor$	<code>\lfloor</code>	$\lceil$	<code>\lceil</code>
$\rangle$	<code>\rangle</code>	$\rfloor$	<code>\rfloor</code>	$\rceil$	<code>\rceil</code>
$\uparrow$	<code>\uparrow</code>	$\Uparrow$	<code>\Uparrow</code>	$\mid$	<code>\mid o \vert</code>
$\downarrow$	<code>\downarrow</code>	$\Downarrow$	<code>\Downarrow</code>	$\parallel$	<code>\parallel o \Vert</code>
$\updownarrow$	<code>\updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>	.	(carattere fantasma)
$/$	<code>/</code>	$\backslash$	<code>\backslash</code>	$\ulcorner$	<code>\ulcorner</code>
$\urcorner$	<code>\urcorner</code>	$\llcorner$	<code>\llcorner</code>	$\lrcorner$	<code>\lrcorner</code>

Tabella 7.11: Grandi delimitatori.

$($	<code>\lgroup</code>	$\int$	<code>\lmoustache</code>	$ $	<code>\arrowvert</code>
$)$	<code>\rgroup</code>	$\backslash$	<code>\rmoustache</code>	$\parallel$	<code>\Arrowvert</code>
$ $	<code>\bracevert</code>				

Tabella 7.12: Frecce.

$\leftarrow$	<code>\leftarrow o \gets</code>	$\longleftarrow$	<code>\longleftarrow</code>
$\rightarrow$	<code>\rightarrow o \to</code>	$\longrightarrow$	<code>\longrightarrow</code>
$\leftrightarrow$	<code>\leftrightharpoonup</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Lleftarrow$	<code>\Lleftarrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Rrightarrow$	<code>\Rrightarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Longleftrightarrow$	<code>\Longleftrightarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>
$\Uparrow$	<code>\Uparrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\Downarrow$	<code>\Downarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\Updownarrow$	<code>\Updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\iff$ (con spazi grandi)	<code>\iff</code> (con spazi grandi)
$\dashleftarrow$	<code>\dashleftarrow</code>	$\dashrightarrow$	<code>\dashrightarrow</code>
$\leftleftarrows$	<code>\leftleftarrows</code>	$\rightrightarrows$	<code>\rightrightarrows</code>
$\leftrightarrows$	<code>\leftrightarrows</code>	$\rightleftarrows$	<code>\rightleftarrows</code>
$\Lleftarrow$	<code>\Lleftarrow</code>	$\Rrightarrow$	<code>\Rrightarrow</code>
$\twoheadleftarrow$	<code>\twoheadleftarrow</code>	$\twoheadrightarrow$	<code>\twoheadrightarrow</code>
$\leftarrowtail$	<code>\leftarrowtail</code>	$\rightarrowtail$	<code>\rightarrowtail</code>
$\leftrightharpoons$	<code>\leftrightharpoons</code>	$\rightleftharpoons$	<code>\rightleftharpoons</code>
$\Lsh$	<code>\Lsh</code>	$\Rsh$	<code>\Rsh</code>
$\looparrowleft$	<code>\looparrowleft</code>	$\looparrowright$	<code>\looparrowright</code>
$\curvearrowleft$	<code>\curvearrowleft</code>	$\curvearrowright$	<code>\curvearrowright</code>
$\circlearrowleft$	<code>\circlearrowleft</code>	$\circlearrowright$	<code>\circlearrowright</code>
$\nearrow$	<code>\nearrow</code>	$\searrow$	<code>\searrow</code>
$\swarrow$	<code>\swarrow</code>	$\nwarrow$	<code>\nwarrow</code>
$\leadsto$	<code>\leadsto</code>	$\multimap$	<code>\multimap</code>
$\upuparrows$	<code>\upuparrows</code>	$\downdownarrows$	<code>\downdownarrows</code>
$\upharpoonleft$	<code>\upharpoonleft</code>	$\upharpoonright$	<code>\upharpoonright</code>
$\downharpoonleft$	<code>\downharpoonleft</code>	$\downharpoonright$	<code>\downharpoonright</code>
$\rightsquigarrow$	<code>\rightsquigarrow</code>	$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>

Tabella 7.13: Simboli misti.

...	<code>\dots</code>	$\Re$	<code>\Re</code>	∞	<code>\infty</code>
...	<code>\cdots</code>	$\Im$	<code>\Im</code>	∂	<code>\partial</code>
$\vdots$	<code>\vdots</code>	$\forall$	<code>\forall</code>	∇	<code>\nabla</code>
$\ddots$	<code>\ddots</code>	$\exists$	<code>\exists</code>	$\emptyset$	<code>\emptyset</code>
$\heartsuit$	<code>\heartsuit</code>	$\neg$	<code>\neg</code> o <code>\lnot</code>	$\complement$	<code>\complement</code>
$\diamondsuit$	<code>\diamondsuit</code>	ℓ	<code>\ell</code>	$\aleph$	<code>\aleph</code>
$\clubsuit$	<code>\clubsuit</code>	$\hbar$	<code>\hbar</code>	$'$	<code>'</code>
$\spadesuit$	<code>\spadesuit</code>	$\hslash$	<code>\hslash</code>	$\prime$	<code>\prime</code>
$\imath$	<code>\imath</code>	$\wp$	<code>\wp</code>	$\mho$	<code>\mho</code>
$\jmath$	<code>\jmath</code>	$\triangle$	<code>\triangle</code>	$\Box$	<code>\Box</code>
$\Diamond$	<code>\Diamond</code>	$\perp$	<code>\bot</code>	$\top$	<code>\top</code>
$\angle$	<code>\angle</code>	$\sqrt{}$	<code>\sqrt</code>	$\flat$	<code>\flat</code>
$\natural$	<code>\natural</code>	$\sharp$	<code>\sharp</code>	$\Bbbk$	<code>\Bbbk</code>
$\square$	<code>\square</code>	$\blacksquare$	<code>\blacksquare</code>	$\textcircled{S}$	<code>\circledS</code>
$\vartriangle$	<code>\vartriangle</code>	$\blacktriangle$	<code>\blacktriangle</code>	$\eth$	<code>\eth</code>
$\triangledown$	<code>\triangledown</code>	$\blacktriangledown$	<code>\blacktriangledown</code>	$\Game$	<code>\Game</code>
$\lozenge$	<code>\lozenge</code>	$\blacklozenge$	<code>\blacklozenge</code>	$\bigstar$	<code>\bigstar</code>
$\angle$	<code>\angle</code>	$\diagup$	<code>\diagup</code>	$\nexists$	<code>\nexists</code>
$\measuredangle$	<code>\measuredangle</code>	$\diagdown$	<code>\diagdown</code>	$\Finv$	<code>\Finv</code>
$\sphericalangle$	<code>\sphericalangle</code>	$\backprime$	<code>\backprime</code>	$\varnothing$	<code>\varnothing</code>

Tabella 7.14: Simboli non matematici (questi simboli possono venire usati anche in modo testuale).

$\dagger$	<code>\dag</code>	$\S$	<code>\S</code>	$\copyright$	<code>\copyright</code>
$\ddagger$	<code>\ddag</code>	$\P$	<code>\P</code>	$\pounds$	<code>\pounds</code>

Tabella 7.15: Altri caratteri alfabetici.

$\digamma$	<code>\digamma</code>	$\varkappa$	<code>\varkappa</code>	$\beth$	<code>\beth</code>
$\daleth$	<code>\daleth</code>	$\gimel$	<code>\gimel</code>		

Tabella 7.16: Altre relazioni binarie.

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

Tabella 7.17: Altre operazioni binarie.

$\dot{+}$	<code>\dotplus</code>	$\cdot$	<code>\centerdot</code>
$\ltimes$	<code>\ltimes</code>	$\rtimes$	<code>\rtimes</code>
$\cup$	<code>\Cup</code> o <code>\doublecup</code>	$\cap$	<code>\Cap</code> o <code>\doublecap</code>
$\veebar$	<code>\veebar</code>	$\barwedge$	<code>\barwedge</code>
$\boxplus$	<code>\boxplus</code>	$\boxminus$	<code>\boxminus</code>
$\boxtimes$	<code>\boxtimes</code>	$\boxdot$	<code>\boxdot</code>
$\leftthreetimes$	<code>\leftthreetimes</code>	$\rightthreetimes$	<code>\rightthreetimes</code>
$\curlyvee$	<code>\curlyvee</code>	$\curlywedge$	<code>\curlywedge</code>
$\intercal$	<code>\intercal</code>	$\divideontimes$	<code>\divideontimes</code>
$\smallsetminus$	<code>\smallsetminus</code>	$\doublebarwedge$	<code>\doublebarwedge</code>
$\circleddash$	<code>\circleddash</code>	$\circledcirc$	<code>\circledcirc</code>
$\circledast$	<code>\circledast</code>		

Tabella 7.18: Negazioni di simboli.

$\nless$	<code>\nless</code>	$\ngtr$	<code>\ngtr</code>	$\nsubsetneqq$	<code>\nsubsetneqq</code>
$\lneq$	<code>\lneq</code>	$\gneq$	<code>\gneq</code>	$\nvarsupsetneqq$	<code>\nvarsupsetneqq</code>
$\nleq$	<code>\nleq</code>	$\ngeq$	<code>\ngeq</code>	$\nsubseteq$	<code>\nsubseteq</code>
$\nleqslant$	<code>\nleqslant</code>	$\ngeqslant$	<code>\ngeqslant</code>	$\nseteq$	<code>\nseteq</code>
$\lneqq$	<code>\lneqq</code>	$\gneqq$	<code>\gneqq</code>	$\nmid$	<code>\nmid</code>
$\lvertneqq$	<code>\lvertneqq</code>	$\gvertneqq$	<code>\gvertneqq</code>	$\nparallel$	<code>\nparallel</code>
$\nleqq$	<code>\nleqq</code>	$\ngeqq$	<code>\ngeqq</code>	$\nshortmid$	<code>\nshortmid</code>
$\lnsim$	<code>\lnsim</code>	$\gnsim$	<code>\gnsim</code>	$\nshortparallel$	<code>\nshortparallel</code>
$\lnapprox$	<code>\lnapprox</code>	$\gnapprox$	<code>\gnapprox</code>	$\nsim$	<code>\nsim</code>
$\nprec$	<code>\nprec</code>	$\nsucc$	<code>\nsucc</code>	$\ncong$	<code>\ncong</code>
$\npreceq$	<code>\npreceq</code>	$\nsucceq$	<code>\nsucceq</code>	$\nvdash$	<code>\nvdash</code>
$\precneqq$	<code>\precneqq</code>	$\succneqq$	<code>\succneqq</code>	$\nvDash$	<code>\nvDash</code>
$\precnsim$	<code>\precnsim</code>	$\succnsim$	<code>\succnsim</code>	$\nVdash$	<code>\nVdash</code>
$\precnapprox$	<code>\precnapprox</code>	$\succnapprox$	<code>\succnapprox</code>	$\nVDash$	<code>\nVDash</code>
$\subsetneq$	<code>\subsetneq</code>	$\supsetneq$	<code>\supsetneq</code>	$\ntriangleleft$	<code>\ntriangleleft</code>
$\varsubsetneq$	<code>\varsubsetneq</code>	$\varsupsetneq$	<code>\varsupsetneq</code>	$\ntriangleright$	<code>\ntriangleright</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\ntrianglelefteq$	<code>\ntrianglelefteq</code>
$\subsetneqq$	<code>\subsetneqq</code>	$\supsetneqq$	<code>\supsetneqq</code>	$\ntrianglerighteq$	<code>\ntrianglerighteq</code>
$\nleftarrow$	<code>\nleftarrow</code>	$\nrightarrow$	<code>\nrightarrow</code>	$\nleftrightarrow$	<code>\nleftrightarrow</code>
$\nLeftarrow$	<code>\nLeftarrow</code>	$\nRightarrow$	<code>\nRightarrow</code>	$\nLeftrightarrow$	<code>\nLeftrightarrow</code>

Tabella 7.19: Font matematici.

Esempio	Comando	Pacchetto richiesto
$ABCdef$	<code>\mathit{ABCdef}</code>	
ABCdef	<code>\mathbf{ABCdef}</code>	
$ABCdef$	<code>\mathrm{ABCdef}</code>	
$ABCdef$	<code>\mathtt{ABCdef}</code>	
$ABCdef$	<code>\mathsf{ABCdef}</code>	
$ABCdef$	<code>\mathnormal{ABCdef}</code>	
ABC	<code>\mathcal{ABC}</code>	
$\mathscr{ABC}$	<code>\mathscr{ABC}</code>	mathrsfs
$\mathcal{ABC}$	<code>\mathcal{ABC}</code>	eucal
$\mathbf{ABC}$	<code>\mathbf{ABC}</code>	amssymb

Capitolo 8

La bibliografia

Realizzare e gestire una bibliografia è generalmente piuttosto complicato. Con $\text{\LaTeX}$ sono disponibili svariati strumenti per svolgere questo lavoro con grande efficienza e flessibilità. Una bibliografia può essere creata manualmente per mezzo dell'ambiente `thebibliography`, oppure in maniera automatizzata mediante il programma `BibTeX`, che permette di separare il contenuto dei riferimenti bibliografici (archiviati in apposite basi di dati) dalla loro forma (gestita da stili).

8.1 L'ambiente `thebibliography`

L'ambiente `thebibliography` permette di gestire la bibliografia in modo molto semplice. Di seguito è riportato un esempio di bibliografia realizzata con l'ambiente `thebibliography`.

```
\begin{thebibliography}{9}
\bibitem{mori:tesi}
L.~F.~Mori (2007),
«Scrivere la tesi di laurea
con  $\text{\LaTeX}$ »,  $\text{\Ars}$ , (2),
\url{http://www.guit.sssup.
it/arstexnica/}.

\bibitem{pantieri:arte}
L.~Pantieri (2008),
\emph{L'arte di scrivere
con  $\text{\LaTeX}$ },
Aracne Editrice, Roma.
\end{thebibliography}
```

Bibliografia

- [1] L. F. Mori (2007), «Scrivere la tesi di laurea con $\text{\LaTeX 2}_{\epsilon}$ », $\text{\ArstExnica}$, (2), <http://www.guit.sssup.it/arstexnica/>.
- [2] L. Pantieri (2008), *L'arte di scrivere con $\text{\LaTeX}$* , Aracne Editrice, Roma.

dove il logo di $\text{\ArstExnica}$ è ottenuto con il comando `\Ars` del pacchetto `guit`.

Per aggiungere all'indice la voce “Bibliografia”, con le classi `book` o `report` è sufficiente scrivere

```
\cleardoublepage
\addcontentsline{toc}{chapter}{\bibname}
```

subito prima di `\begin{thebibliography}`. In alternativa, se si sta usando la classe `article`, il comando

`\addcontentsline{toc}{section}{\refname}`

(anch'esso da scriversi subito prima di `\begin{thebibliography}` ed eventualmente preceduto da `\clearpage` se la bibliografia inizia in una nuova pagina) aggiunge all'indice la voce "Riferimenti bibliografici".

Il secondo argomento del comando `\addcontentsline` precisa il livello del titolo da aggiungere all'indice (`toc`), `chapter` per le classi `book` o `report`, `section` per la classe `article`.

Il comando `\cleardoublepage` prima di `\addcontentsline` nella bibliografia delle classi `report` o `book` (come il comando `\clearpage` nella bibliografia della classe `article` quando questa inizia in una nuova pagina) è necessario: in caso contrario, il numero di pagina che compare nell'indice non è corretto.

L'ambiente `thebibliography` è simile a `itemize`. Ogni elemento della bibliografia comincia con il comando `\bibitem`, che ha come argomento l'etichetta che identifica (come con `\label`) il documento in questione; di seguito si scrivono l'autore dell'opera, il titolo (in corsivo, nel caso considerato), l'editore e l'anno di pubblicazione. Gli elementi della bibliografia vanno ordinati a mano, poiché `thebibliography` (contrariamente a `BiBTeX`) non provvede a farlo automaticamente.

In stampa, ogni elemento della bibliografia viene contrassegnato da un numero tra parentesi quadre. È possibile assegnare a ciascuna voce bibliografica un'etichetta personalizzata con il parametro opzionale di `\bibitem`:

`\bibitem[Pantieri, 09]{pantieri:arte} L.~Pantieri ...`

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

- se non si usano etichette personalizzate, in genere si mette 9 nel caso in cui ci siano meno di dieci opere in bibliografia, 99 qualora se ne abbiano almeno dieci ma meno di cento, ...;
- se si usano etichette personalizzate, si mette l'etichetta più lunga.

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

Vedi `\cite{mori:tesi}` per
maggiori dettagli.

Vedi [1] per maggiori dettagli.

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

Vedi `\cite[pag.~27]{mori:tesi}`
per maggiori dettagli.

Vedi [1, pag. 27] per maggiori dettagli.

Capitolo 9

L'indice analitico

L'indice analitico costituisce, in molti lavori, una caratteristica di grande utilità. Come la bibliografia, anche l'indice analitico può essere gestito automaticamente da $\text{\LaTeX}$. In questo capitolo vengono illustrate le nozioni essenziali, veramente semplici, per generare l'indice analitico.

9.1 Creare l'indice analitico

Per abilitare la composizione dell'indice analitico da parte di $\text{\LaTeX}$, si deve richiamare nel preambolo il pacchetto `makeidx`,

```
\usepackage{makeidx}
```

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

```
\makeindex
```

inserito anch'esso nel preambolo del file sorgente. È inoltre necessario scrivere il comando

```
\printindex
```

nel punto in cui si desidera che compaia l'indice analitico (tipicamente, alla fine del documento, subito prima di `\end{document}`).

Il contenuto dell'indice analitico viene specificato attraverso il comando

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

dove *<voce>* indica la voce dell'indice analitico; tale comando va inserito nel punto del testo dove si desidera che si riferisca alla voce presente nell'indice analitico. Il codice

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

La parola “arte” è nell'indice analitico.

genera una voce per “arte” nell'indice analitico.

Si può mettere tutta un'espressione nell'indice analitico:

```
È possibile mettere tutta  
un'espressione\index{tutta  
un'espressione} nell'indice  
analitico.
```

È possibile mettere tutta un'espressione
nell'indice analitico.

Tabella 9.1: Esempi di voci dell'indice analitico.

Esempio	Voce
Voce primaria <code>\index{Arte}</code>	Arte, 2
Voce secondaria sotto "Arte" <code>\index{Arte!Escher}</code>	Escher, 3
Voce con forma specificata <code>\index{Gaudì@\textit{Gaudì}}</code>	<i>Gaudì</i> , 5
Pagina con forma specificata <code>\index{Klimt textbf}</code>	Klimt, 7

Per generare dei sottolemmi bisogna separare il lemma principale e il sottolemma con il carattere !:

Qui parliamo dei sottolemmi%
`\index{indice!sottolemmi}`
 dell'indice analitico.

Qui parliamo dei sottolemmi dell'indice
 analitico.

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

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

Mettiamo il simbolo α nell'indice ana-
 litico al posto di "alpha" nell'ordine
 alfabetico.

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

È possibile inserire una voce in
 corsivo`\index{corsivo@%`
`\textit{corsivo}}}`.

È possibile inserire una voce in corsivo.

La tabella 9.1 illustra, per mezzo di alcuni esempi, la sintassi del comando `\index`.

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

1. si compila una prima volta con $\text{\LaTeX}$ il sorgente;
2. si compila con il programma `MakeIndex`;
3. si compila altre due volte con $\text{\LaTeX}$.

Quando $\text{\LaTeX}$ viene eseguito la prima volta sul file sorgente, ogni comando `\index` scrive un'opportuna voce insieme al numero di pagina corrispondente in un file speciale che ha lo stesso nome del file sorgente $\text{\LaTeX}$, ma estensione diversa (`.idx`). Questo file `.idx` viene poi elaborato dal programma `MakeIndex`, che genera un file avente ancora lo stesso nome, ma questa volta estensione `.ind`,

che contiene le voci nella forma “giusta”, cioè in ordine alfabetico, strutturate per lemmi, sottolemmi e sotto-sottolemmi, con gli elenchi di numeri di pagine senza ripetizioni ed eventualmente con intervalli, se necessario composti con caratteri diversi a seconda dell’uso della voce. Se a questo punto si compila ancora due volte il sorgente $\text{\LaTeX}$, questo indice analitico ordinato verrà incluso nel documento e verranno aggiornati tutti i riferimenti.

In fase di correzione di bozze e controllo dell’indice analitico può essere utile il pacchetto `showidx`, che stampa le voci dell’indice analitico nel margine sinistro della pagina.

9.2 Bilanciare le colonne dell’indice analitico

Per bilanciare le colonne dell’ultima pagina dell’indice analitico è sufficiente inserirlo all’interno di un ambiente `multicols` (deve essere caricato il pacchetto `multicol`). A questo scopo è possibile ridefinire l’ambiente `theindex` con il seguente codice,

```
\let\orgtheindex\theindex
\let\orgendtheindex\endtheindex
\def\theindex{%
    \def\twocolumn{\begin{multicols}{2}}%
    \def\onecolumn{}%
    \clearpage
    \orgtheindex
}
\def\endtheindex{%
    \end{multicols}%
    \orgendtheindex
}
```

da scrivere nel preambolo.

Capitolo 10

Personalizzare L^AT_EX

La ricerca di personalizzazioni di particolari oggetti o dell'intero documento è un processo che presto o tardi tutti si trovano ad affrontare. In questo capitolo vengono esposti alcuni suggerimenti su come fare in modo che L^AT_EX produca risultati diversi da quelli predefiniti.

10.1 Comandi, ambienti e pacchetti nuovi

10.1.1 Definire nuovi comandi

Uno degli aspetti più importanti della filosofia di L^AT_EX è la distinzione tra la struttura del documento e la sua “resa” visiva. Si immagini per esempio di dover scrivere un libro di botanica e di volere che tutti i nomi latini delle piante appaiano in corsivo. In teoria, basterebbe metterli come argomento del comando `\textit`. Se però l'editore dovesse chiedere di comporre i nomi latini in neretto e non in corsivo (perché queste sono le sue scelte tipografiche), sarebbe necessario sostituire tutti i `\textit` in `\textbf`, facendo attenzione, perché probabilmente il corsivo sarà stato usato anche per qualcos'altro. Tutto ciò non è il massimo della praticità.

Conformemente alla sua filosofia, L^AT_EX offre una soluzione elegante a questo problema. È possibile infatti definire un nuovo comando, `\latino`, che stampa il suo argomento, cioè un nome latino, nello stile specificato. Se i nomi latini devono poi essere in neretto, basterà cambiare la definizione di `\latino`. Il comando `\latino`, dunque, non descrive il modo in cui il suo argomento deve essere *stampato*, ma il suo valore, il modo in cui lo si è *pensato* nell'ambito del testo. Ecco perché non si usano quasi mai dei comandi di cambiamento di stile in un documento scritto con L^AT_EX: si preferisce definire un insieme di comandi che rispecchiano la logica del testo e servirsi di questi ultimi.

Per aggiungere comandi personali, si usa il comando

```
\newcommand{⟨nome⟩}[⟨numero di argomenti⟩][⟨predefinito⟩]{⟨definizione⟩}
```

I due argomenti obbligatori sono il `⟨nome⟩` del nuovo comando, e la `⟨definizione⟩` del comando. Il parametro opzionale `⟨numero di argomenti⟩` specifica il numero di argomenti che il nuovo comando richiede (fino a un massimo di nove); se manca, si assume che non vi sia alcun argomento. Il primo degli argomenti del nuovo comando può essere facoltativo; esso è facoltativo se viene espressa anche la seconda

opzione di `\newcommand` (si può dare anche solo []). Se nell'uso di un nuovo comando che accetta un argomento facoltativo quest'ultimo non viene specificato, esso assume il valore *predefinito*.

Per esempio, è possibile definire un nuovo comando, chiamato `\impaziente`, che è l'abbreviazione di L^AT_EX per l'impaziente, nel modo seguente:

```
\newcommand{\impaziente}{\emph{\LaTeX} per l'impaziente}}
```

Un comando simile potrebbe essere utile se si dovesse scrivere ripetutamente il titolo di questo libro.

Questo lavoro è intitolato
`\impaziente`.

Questo lavoro è intitolato L^AT_EX per
l'impaziente.

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

```
\newcommand{\latino}[1]{\textit{#1}}
```

il comando `\latino` si usa nel modo seguente:

`\latino{rosa canina}`

rosa canina

Se si desidera creare nuovi comandi che sovrascrivano quelli già esistenti, è necessario usare il comando `\renewcommand`; la sintassi è la stessa di `\newcommand`.

10.1.2 Nuovi ambienti

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

```
\newenvironment{<nome>}[<numero di argomenti>][<predefinito>]{<prima>}{<dopo>}
```

Quando L^AT_EX incontra `\begin{<nome>}`, lo sostituisce con quanto specificato nell'argomento *prima*, e quando incontra `\end{<nome>}`, lo sostituisce con quanto specificato nell'argomento *dopo*. L'ambiente che viene definito può ricevere (in apertura) un certo *numero di argomenti*, il primo dei quali può essere facoltativo, se viene specificato il valore *predefinito*, e va trattato come il primo argomento facoltativo definito con `\newcommand`.

Il seguente esempio mostra l'uso del comando `\newenvironment`. Se scriviamo nel preambolo

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

l'ambiente `itaitemize` si usa nel modo seguente:

```
\begin{itaitemize}
```

```
\item Un elenco con voci\dots
```

```
\item \dots automaticamente
```

```
in corsivo.
```

```
\end{itaitemize}
```

• *Un elenco con voci...*

• *... automaticamente in corsivo.*

Il parametro opzionale *numero di argomenti* si usa come nel comando `\newcommand`.

L^AT_EX si assicura che non venga definito un ambiente già esistente: se si vuole ridefinire un ambiente già esistente, si usa il comando `\renewenvironment`, che ha la stessa sintassi di `\newenvironment`.

Tabella 10.1: Comandi per cambiare lo stile dei font.

Comando	Dichiarazione	Stile
<code>\textit</code>	<code>\itshape</code>	<i>Corsivo</i>
<code>\textsc</code>	<code>\scshape</code>	MAIUSCOLETTO
<code>\textbf</code>	<code>\bfseries</code>	Neretto
<code>\textsl</code>	<code>\slshape</code>	<i>Inclinato</i>
<code>\textrm</code>	<code>\rmfamily</code>	Tondo
<code>\textsf</code>	<code>\sffamily</code>	Senza grazie
<code>\texttt</code>	<code>\ttfamily</code>	Macchina per scrivere

10.1.3 Un pacchetto personale

Quando vengono definiti molti nuovi comandi e ambienti, il preambolo del documento può diventare piuttosto lungo. In questo caso, è una buona prassi creare un pacchetto contenente le definizioni dei propri comandi e ambienti; si usa poi il comando `\usepackage` per caricare il pacchetto nel documento principale.

```
% Un esempio di pacchetto personale
\ProvidesPackage{esempio}
\newcommand{\impaziente}{\emph{\LaTeX{}} per l'impaziente}}
\newcommand{\latino}[1]{\textit{#1}}
\newenvironment{itaitemize}{\begin{itemize}\itshape}{\end{itemize}}
```

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

```
\ProvidesPackage{<nome del pacchetto>}
```

da usare all'inizio del file contenente pacchetto. Questo comando comunica a $\text{\LaTeX}$ il nome del pacchetto, permettendogli di fornire messaggi di errore appropriati se si tenta di includere due volte lo stesso pacchetto.

10.2 Font

Di regola, $\text{\LaTeX}$ sceglie il font appropriato in base alla struttura logica del documento (capitoli, paragrafi, testatine, ...). In alcuni casi, si potrebbe però voler cambiare lo stile e il corpo dei font, manualmente.

10.2.1 Comandi per cambiare lo stile dei font

Per cambiare lo stile dei font, si usano i comandi elencati nella tabella 10.1. Questi comandi modificano unicamente lo stile del loro argomento, mentre il testo che segue non viene modificato.

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

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

È possibile combinare i comandi:

Tabella 10.2: Dichiarazioni per cambiare il corpo dei font.

Dichiarazione	Corpo
<code>\tiny</code>	Font minuscolo
<code>\scriptsize</code>	Font molto piccolo
<code>\footnotesize</code>	Font abbastanza piccolo
<code>\small</code>	Font piccolo
<code>\normalsize</code>	Font normale
<code>\large</code>	Font grande
<code>\Large</code>	Font più grande
<code>\LARGE</code>	Font molto grande
<code>\huge</code>	Font enorme
<code>\Huge</code>	Il più grande

L'espressione che segue
`\textit{è in \textbf{neretto corsivo}}`.

L'espressione che segue è in **neretto corsivo**.

Per ciascuno di questi comandi, esiste una *dichiarazione* corrispondente, ovvero un comando privo di argomenti, che agisce su tutto il resto del testo. Per limitare l'effetto di una dichiarazione bisogna usare delle *graffe di raggruppamento*.

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

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

Si possono anche combinare più dichiarazioni:

L'espressione che segue
`{\itshape è in {\bfseries neretto corsivo}}`.

L'espressione che segue è in **neretto corsivo**.

Gli spazi che seguono una dichiarazione vengono ignorati.

10.2.2 Dichiarazioni per cambiare il corpo dei font

Per cambiare il corpo dei font, si usano le dichiarazioni elencate nella tabella 10.2.

Lettere `{\Large grandi e {\footnotesize piccole}}`.

Lettere **grandi** e piccole.

Il corpo effettivo di ogni font dipende dalla classe del documento e dalle sue opzioni; nella tabella 10.3 nella pagina successiva sono elencati i corpi, in punti, dei font corrispondenti alle diverse dichiarazioni nelle classi standard, in funzione dell'opzione di classe selezionata.

Le dichiarazioni per il cambiamento della dimensione cambiano anche l'interlinea, ma solo se il capoverso termina entro il raggio d'azione del comando. Si confronti la posizione del comando `\par` nei due seguenti esempi.

Tabella 10.3: I corpi dei font nelle classi standard (il corpo predefinito è di 10 punti).

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

```
{\large Socrate: «Platone
mentirà nella frase
seguente».\par}
```

Socrate: «Platone mentirà nella frase seguente».

```
{\large Platone: «Socrate ha
detto il vero nella frase
precedente».\par}
```

Platone: «Socrate ha detto il vero nella frase precedente».

Nel secondo esempio viene un'interlinea scorretta: il comando `\par` deve essere presente *prima* della parentesi graffa chiusa.

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

Ricorda! *Tanti Più font scegli* di usare in un documento, *quanto più* ILLEGGIBILE e *brutto diventa*.

10.3 Le pagine bianche nei documenti fronte-retro

Se si sta scrivendo un libro o una relazione, l'opzione `openright`, predefinita nella classe `book`, fa iniziare i capitoli sempre in una pagina destra (vedi il paragrafo 4.4 a pagina 22). Se si usano le classi standard, per far sì che le eventuali pagine bianche inserite alla fine di un capitolo non abbiano né testatina né piè di pagina (scelta peraltro consigliabile), è sufficiente caricare il pacchetto `emptypage`:

```
\usepackage{emptypage}
```


Appendice A

Norme tipografiche italiane

Questa appendice, basata su [5], cui si rimanda per ogni approfondimento, descrive sinteticamente le principali norme tipografiche della lingua italiana, utili nella composizione di articoli, tesi o libri. Per ogni regola discussa, si mostra come applicarla in $\text{\LaTeX}$.

Va tenuto presente che la scrittura in una qualunque lingua è regolata nel mondo dalle norme emesse dall'iso, in Italia dalle norme definite dall'UNI, che (come detto a pagina 77) hanno valore di legge. Per maggiori dettagli sulle norme ISO-UNI si rimanda alla relativa documentazione e a [2].

In questa appendice diamo per scontato che siano caricati correttamente il pacchetto `babel` con l'opzione `italian`, il pacchetto `inputenc` con la codifica appropriata e il pacchetto `fontenc` con l'opzione `T1` (vedi i primi due paragrafi del capitolo 5 a pagina 33):

```
\usepackage[italian]{babel}
\usepackage[<codifica>]{inputenc}
\usepackage[T1]{fontenc}
```

A.1 L'accento e l'apostrofo

A.1.1 Uso dell'accento

In italiano, l'accento assume due forme diverse, chiamate rispettivamente *accento grave* (da sinistra in alto a destra in basso: `), che segnala un suono aperto della vocale, e *accento acuto* (da sinistra in basso a destra in alto: ´), che segnala un suono chiuso.

Gli accenti obbligatori sono sempre gravi su tutte le vocali tranne sulla *e* che può riceverlo sia grave sia acuto a seconda dei casi.

I monosillabi che devono essere accentati sono i seguenti: *ché* (congiunzione causale), *chiù* (onomatopea), *ciò* (pronome), *dà* (indicativo di dare), *diè* (passato remoto di dare), *dì* (giorno), *è* (indicativo di essere), *fè* (fede), *già* (avverbio), *giù* (avverbio), *là* (avverbio), *lì* (avverbio), *né* (congiunzione), *piè* (piede), *più* (avverbio), *può* (indicativo di potere), *sé* (pronome tonico), *scià* (titolo nobiliare), *sì* (avverbio), *tè* (bevanda).

Per sapere quale accento indicare sulla lettera *e* bisognerebbe fare riferimento alla pronuncia aperta o chiusa, ma siccome questa è molto variabile da persona a persona, le norme UNI stabiliscono che:

- l'accento è grave su *ahimè, ohimè, caffè, canapè, cioè, coccodè, diè, è, gilè, lacchè, piè, tè*; sui francesismi adattati alla grafia italiana come *bebè, cabarè, relè, purè*, eccetera; sui nomi propri *Giosuè, Mosè, Noè, Salomè*, eccetera;
- l'accento è acuto su *ché* e i suoi composti come *perché, affinché*, eccetera; *fé* e i suoi composti come *autodafé*; sui composti di *re* come *viceré*, e sui composti di *tre* come *ventitré, trentatré*; sui passati remoti come *credé, temé*; su *mercé, né, scimpanzé, sé, testé*.

Si noti che, al contrario delle vecchie macchine per scrivere o di altri programmi di composizione del testo, L^AT_EX permette di accentare senza problemi le lettere maiuscole (il codice `\`E` produce direttamente “È”), che spesso vengono erroneamente rese con un apostrofo (“E’”, che significa “ei, egli”).

È possibile usare il pacchetto `inputenc` per inserire i caratteri accentati direttamente da tastiera (vedi il paragrafo 5.1.2 a pagina 35).

A.1.2 Apostrofo

L'*apostrofo*, `'`, segnala normalmente la caduta della parte iniziale, come in *'sta* (per “questa”), o finale di una parola, come in *un'altra, un po', da'* (imperativo), eccetera.

Quando cadono lettere o sillabe iniziali l'apostrofo è preceduto da uno spazio. Quando cadono *sillabe* finali l'apostrofo è seguito da uno spazio o da un segno di interpunzione.

Quando in una parola cade una vocale finale di fronte a un'altra parola che inizia per vocale (come in *un'altra* o in *quell'uomo*), l'apostrofo sta al posto della vocale elisa; in quest'ultimo caso nel file sorgente non bisogna mai lasciare uno spazio dopo l'apostrofo. In questo modo, L^AT_EX evita automaticamente che resti l'apostrofo in fin di riga. Per esempio, “quell'uomo” viene diviso in sillabe come una parola unica in “quel-l'uo-mo” (occorre l'opzione `italian` di `babel`).

A.2 Punteggiatura e spaziatura

A.2.1 Segni di interpunzione

Tutti i segni di interpunzione (punto, virgola, punto e virgola, due punti, punti interrogativo ed esclamativo, ...) vanno attaccati alla parola che precede e separati per mezzo di uno spazio da quella che segue. Quindi, in generale, dopo ogni segno di interpunzione va battuto uno spazio.

Uso di virgolette e parentesi

Sia le virgolette (vedi il paragrafo 5.7.1 a pagina 45) sia le parentesi vanno attaccate alle parole che racchiudono, e separate con uno spazio, sia prima sia dopo, dal resto della frase (a meno che non siano a fine frase, come qui). Quando un segno

d'interpunzione ricorre dentro a virgolette o a parentesi, la frase o il periodo finirà, come questo, con un punto *fuori* dalla parentesi stessa (ovviamente!). Si noti che solo punto interrogativo e punto esclamativo stanno di norma dentro alle parentesi.

A.2.2 Punti ellittici

I punti ellittici sono sempre e solo tre, e, come gli altri segni di interpunzione, sono attaccati alla parola che precede e separati con uno spazio da quella che segue... in questo modo. In $\text{\LaTeX}$, il comando da usare è `\dots` (non vanno mai inseriti a mano tre punti separati).

Se usati per indicare un'omissione in una citazione «è bene [...] inserire i puntini entro parentesi quadre o tonde» come in questo caso.

A.3 Stile dei font

A.3.1 Corsivo

Il corsivo si usa nel testo principalmente:

- per evidenziare le parole *importanti* del testo;
- per le *parole straniere* (anche latine) non entrate nell'uso italiano, per esempio *guillemet* o *politically correct*;
- per i titoli di opere citate, per esempio *La Divina Commedia* o *The $\text{\TeX}$ book*;

In bibliografia, si usa per i titoli dei libri e delle collezioni, ma questo dipende dallo stile bibliografico scelto.

A.3.2 Neretto

Il neretto si usa, nelle classi standard di $\text{\LaTeX}$, per i titoli dei capitoli, dei paragrafi e delle altre suddivisioni del testo. Si tende di norma a non impiegarlo per evidenziare le parole, uso per cui esiste già il corsivo, o comunque a utilizzarlo con moderazione, per non appesantire l'aspetto della pagina.

A.4 Capoversi

Un capoverso è un blocco di testo, contenente uno o più periodi, che logicamente individua una parte del discorso autonoma.

Di regola, la prima riga di un capoverso viene rientrata per evidenziare lo stacco da quello precedente. In questo caso, in italiano viene solitamente rientrata anche quella del primo capoverso di un capitolo o di un paragrafo (mentre in inglese solo il secondo capoverso viene rientrato).

Per avere il rientro anche nel paragrafo iniziale si usa il pacchetto `indentfirst`, richiamandolo nel preambolo con:

```
\usepackage{indentfirst}
```


Bibliografia

- [1] AMERICAN MATHEMATICAL SOCIETY (1999), *Manuale d'uso del pacchetto amsmath*, <http://tug.ctan.org/tex-archive/info/italian/amslldoc/itamslldoc.pdf>.
- [2] BECCARI, C. (2009), *Introduzione all'arte della composizione tipografica con L^AT_EX*, <http://www.guit.sssup.it/downloads/GuidaGuIT.pdf>.
- [3] BUSDRAGHI, F. (2004), *Documentazione del pacchetto layaureo*, <http://www.guit.sssup.it/downloads/layaureo.pdf>.
- [4] CAUCCI, L. e SPADACCINI, M. (2000), *Gestione di figure e tabelle con L^AT_EX*, http://www.guit.sssup.it/downloads/fig_tut.pdf.
- [5] CEVOLANI, G. (2006), «Norme tipografiche per l'italiano in L^AT_EX», *Ar^ST_EXⁿica*, (1), <http://www.guit.sssup.it/arstexnica/>.
- [6] FAIRBAIRNS, R. (2007), *The UK T_EX FAQ*, <http://www.tex.ac.uk/tex-archive/help/uk-tex-faq/newfaq.pdf>.
- [7] FEAR, S. (2005), *Publication quality tables in L^AT_EX*, *Manuale d'uso del pacchetto booktabs*, <http://www.ctan.org/tex-archive/macros/latex/contrib/booktabs/booktabs.pdf>.
- [8] GOOSSENS, M., MITTELBACH, F. e BRAAMS, J. (2004), *The L^AT_EX Companion*, Addison-Wesley, Reading (Massachusetts).
- [9] GREGORIO, E. (2003), *Galleria degli orrori: Come maltrattare L^AT_EX e rendere infelice un copy editor*, <http://profs.sci.univr.it/~gregorio/orrori.pdf>.
- [10] GREGORIO, E. (2008), *Appunti di programmazione in L^AT_EX e T_EX*, <http://profs.sci.univr.it/~gregorio/introtex.pdf>.
- [11] GREGORIO, E. (2008), *L^AT_EX: breve guida ai pacchetti di uso più comune*, <http://profs.sci.univr.it/~gregorio/breveguida.pdf>.
- [12] GUIGGIANI, M. e MORI, L. F. (2008), «Consigli su come *non* maltrattare le formule matematiche», *Ar^ST_EXⁿica*, (5), <http://www.guit.sssup.it/arstexnica/>.
- [13] KNUTH, D. E. (1984), *The T_EXbook*, Addison-Wesley, Reading (Massachusetts).

- [14] LAMPORT, L. (1994), *LaTeX: a document preparation system*, Addison-Wesley, Reading (Massachusetts).
- [15] MORI, L. F. (2006), «Tabelle su LaTeX 2_ε: pacchetti e metodi da utilizzare», *ArTeXnica*, (2), <http://www.guit.sssup.it/arstexnica/>.
- [16] MORI, L. F. (2007), «Scrivere la tesi di laurea con LaTeX 2_ε», *ArTeXnica*, (3), <http://www.guit.sssup.it/arstexnica/>.
- [17] MORI, L. F. e HIMMELMANN, M. (2007), «Scrivere il curriculum vitae», *ArTeXnica*, (4), <http://www.guit.sssup.it/arstexnica/>.
- [18] OETIKER, T., PARTL, H., HYNÄ, I. e SCHLEGL, E. (2000), *Una (mica tanto) breve introduzione a LaTeX 2_ε*, <http://www.ctan.org/tex-archive/info/italian/lshort/itlshort.pdf> (è disponibile anche la versione in inglese, aggiornata al 2008: <ftp://tug.ctan.org/pub/tex-archive/info/lshort/english/lshort.pdf>).
- [19] PAKIN, S. (2008), *The Comprehensive LaTeX Symbol List*, <http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>.
- [20] PANTIERI, L. (2008), *Introduzione allo stile ClassicThesis*, http://www.lorenzopantieri.net/LaTeX_files/ClassicThesis.pdf.
- [21] PANTIERI, L. (2009), *L'arte di scrivere con LaTeX*, http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf.
- [22] TRETTIN, M. e ZANNARINI, E. (2005), *Elenco dei "peccati" degli utenti di LaTeX 2_ε*, <ftp://ftp.dante.de/tex-archive/info/l2tabu/italian/l2tabuit.pdf>.
- [23] ZANNARINI, E. e VAVASSORI, E. G. (2005), *LaTeX e i font: installazione pratica*, <http://www.guit.sssup.it/guitmeeting/2005/articoli/zannarini-vavassori.pdf>.

Indice analitico

a4paper, 23	\Big, 84
a5paper, 23	\big, 84
abstract, 24	\Bigg, 84
\addcontentsline, 104	\bigg, 84
align, 87–89	\Biggl, 84
align*, 89	\biggl, 84
aligned, ix, 89	\Biggr, 84
\alpha, 79	\biggr, 84
\AmS, 46	\Bigl, 84
amscd, 95	\bgl, 84
amsmath, 28, 46, 75, 83, 87, 119	\Bigr, 84
amssymb, 28, 75	\bigr, 84
amsthm, 28, 92, 93	\bigskip, 38
\and, 48	binding, 42
ansinew, 35	\binom, 85
\ap, 91	bm, 86, 91
\appendix, 43	\bm, 86, 91
applemac, 35	Bmatrix, 86
\arccos, 84	bmatrix, 86
\arcsin, 84	\bmod, 83
\arcsinh, 83	book, 22–24, 29, 43, 45, 103, 104, 113
\arctan, 84	booktabs, 28, 57, 58, 119
\arg, 84	\bottomrule, 58
array, 59	“bounding box”, 64
\Ars, 20, 46, 103	\Bra, 85
article, 22–24, 29, 43, 103, 104	\Braket, 85
\author, 48	braket, 85
b5paper, 23	caption, 28, 61, 72, 73
babel, 25, 27, 28, 33, 34, 40, 45, 91, 115, 116	\caption, 61, 70, 73
\backmatter, 43	cases, 89
\bar, 80	CD, 95
beamer, 22	center, 52, 61, 67, 70
\beta, 79	\centering, 61, 70
\bfseries, 111	\chapter, 34, 43, 45
\bibitem, 104	\chapter*, 44
	chngepage, 28, 42

- \cite, 28, 104
- Classe
 - article, 22–24, 29, 43, 103, 104
 - beamer, 22
 - book, 22–24, 29, 43, 45, 103, 104, 113
 - examdesign, 22
 - KOMA-Script, 22
 - letter, 22
 - memoir, 22
 - report, 22–24, 29, 43, 103, 104
 - toptesi, 22
- \cleardoublepage, 72, 104
- \clearpage, 104
- Codifica dei font, 28, 35, 36, 115
- \colon, 81
- \columnwidth, 60, 68
 - comment, 20
- \cos, 83, 84
- \cosh, 84
- \cot, 84
- \coth, 84
- cp850, 35
- \csc, 84
- \date, 48
 - dcolumn, 60
- \ddots, 87
- \DeclareMathOperator, 83
 - definition, 93
- \deg, 84
- \Delta, 79
- \delta, 79
 - description, 51
- \det, 84
- \dfrac, 92
- \dim, 84
- \displaystyle, 91
- \documentclass, 22, 30, 39
- \dots, 22, 46, 81, 117
- \doublespacing, 43
 - draft, 23, 39
- \emph, 20
 - empty, 29
 - emptypage, 113
- \enlargethispage, 56
 - enumerate, 51
 - enumitem, 52
- \epsilon, 80
- \eqarray, 87
- \eqref, 76
- equation, 76, 88, 89, 95
- equation*, 76, 88
- \euro, 47
- eurosym, 28, 47
- examdesign, 22
- executivepaper, 23
- \exp, 84
- fancyhdr, 29
- figure, 66, 68–70, 72
- figure*, 68
- final, 23
- fleqn, 23, 88
- float, 28, 72
- \FloatBarrier, 72
- flushleft, 52
- flushright, 52
- font, 73
- fontenc, 28, 36, 115
- \footnotesize, 112, 113
- \foreignlanguage, 34
 - format, 73
- \frac, 78
- \frenchspacing, 41
- frontespizio, 48
- \frontmatter, 43, 45
- \Gamma, 79
- \gamma, 79
- gather, 87–89
- gather*, 88
- gathered, ix, 89
- \gcd, 84
- geometry, 25, 28, 42
- \graphicspath, 66
- graphicx, 13, 23, 25, 28, 64, 65
- \GuIT, 46
 - guit, 20, 24, 46, 103
- \hdotsfor, 87
- headings, 29
- height, 65
- \hline, 58
- \hom, 84

- \Huge, 112, 113
- \huge, 112, 113
 - hyperref, 13, 28, 30, 46
- \hyphenation, 39
- \idotsint, 78
- \iff, 81
- \iiiint, 78
- \iiint, 78
- \iint, 78
- \impaziente, 110
- \implies, 81
- \in, 79
- \includegraphics, 65, 66
 - indentfirst, 28, 41, 53, 117
- \index, 106
- \inf, 84
- \infty, 78
- \input, 31
 - inputenc, 28, 34, 35, 45, 115, 116
- \int, 78
 - itaitemize, 110
 - italian, 40, 45, 91, 115, 116
- \item, 51, 52
 - itemize, 51, 104
- \itshape, 111
- \ker, 84
- \Ket, 85
- KOMA-Script, 22
- \label, 28, 49, 61, 70, 73, 76, 104
- \land, 81
- \LARGE, 112, 113
- \Large, 112, 113
- \large, 112, 113
- \LaTeX, 17, 46
- \latex, 17
- \LaTeXe, 46
 - latin1, 35
 - latin9, 35
- \latino, 109, 110
 - layaureo, 25, 42, 119
- \left, 84
 - legalpaper, 23
 - leqno, 23
 - letter, 22
 - letterpaper, 23
- \lg, 84
- \lim, 84
- \liminf, 84
- \limsup, 84
- \linewidth, 68
 - listings, 28, 55
- \listoffigures, 73
- \listoftables, 61, 73
- \ln, 84
- \lnot, 81
- \log, 83, 84
- \looseness, 55, 56
- \lor, 81
- \lVert, 81
- \lvert, 80
- \mainmatter, 43, 45
 - makeidx, 28, 105
 - MakeIndex, 106
- \maketitle, 22, 48, 50
- \mapsto, 81
 - Margini di pagina, 24, 25, 28, 41, 42, 53
- \markboth, 29, 44
- \markright, 29
- \mathbb, 79
- \mathbf, 86, 90, 91
- \mathcal, 90
 - matrix, 86
- \max, 84
- \medskip, 38, 61
 - memoir, 22
 - microtype, 13, 28, 41
- \mid, 80
- \midrule, 58
- \min, 84
- \mp, 77
 - multicol, 28, 107
 - multicols, 107
- \multicolumn, 59
- \multline, 87, 88
- \multline*, 88
- \myheadings, 29
- \newcommand, 110
- \newenvironment, 110
- \newline, 39
- \newpage, 56
- \newtheorem, 92

- `\newtheoremstyle`, 93
- `\normalsize`, 112, 113
- `\notag`, 88, 89
- `\notin`, 79
- `notitlepage`, 23
- `\oddsidemargin`, 42
- `\onehalfspacing`, 43
- `oneside`, 23, 29
- `openany`, 23
- `openright`, 23, 113
- `\overbrace`, 85
- `\overline`, 80
- `\overrightarrow`, 80
- `\overset`, 80
- Pacchetto
 - `amscd`, 95
 - `amsmath`, 28, 46, 75, 83, 87, 119
 - `amssymb`, 28, 75
 - `amsthm`, 28, 92, 93
 - `array`, 59
 - `babel`, 25, 27, 28, 33, 34, 40, 45, 91, 115, 116
 - `bm`, 86, 91
 - `booktabs`, 28, 57, 58, 119
 - `braket`, 85
 - `caption`, 28, 61, 72, 73
 - `chngpage`, 28, 42
 - `comment`, 20
 - `dcolumn`, 60
 - `emptypage`, 113
 - `enumitem`, 52
 - `eurosym`, 28, 47
 - `fancyhdr`, 29
 - `float`, 28, 72
 - `fontenc`, 28, 36, 115
 - `frontespizio`, 48
 - `geometry`, 25, 28, 42
 - `graphicx`, 13, 23, 25, 28, 64, 65
 - `guit`, 20, 24, 46, 103
 - `hyperref`, 13, 28, 30, 46
 - `indentfirst`, 28, 41, 53, 117
 - `inputenc`, 28, 34, 35, 45, 115, 116
 - `layaureo`, 25, 42, 119
 - `listings`, 28, 55
 - `makeidx`, 28, 105
 - `microtype`, 13, 28, 41
 - `multicol`, 28, 107
 - `placeins`, 72
 - `PSTricks`, 95
 - `setspace`, 43
 - `showidx`, 107
 - `showlabels`, 28
 - `subfig`, 28, 73
 - `subfigure`, 73
 - `syntonly`, 31
 - `tabularx`, 28, 59, 60
 - `ucs`, 35
 - `url`, 28, 46
 - `varioref`, 28, 49
 - `xcolor`, 28
 - `Xy-pic`, 95
- `\pagebreak`, 56
- `\pageref`, 49
- `\par`, 37, 112, 113
- `\paragraph`, 43
- `\parallel`, 81
- `\part`, 43
- `\ped`, 91
- `\phi`, 80
- `placeins`, 72
- `plain`, 29, 93
- `\pm`, 77
- `pmatrix`, 86
- `\pmod`, 83
- `\Pr`, 84
- `\prod`, 78
- `proof`, 94
- `PSTricks`, 95
- `\qqquad`, 76, 82
- `\quad`, 76, 82
- `\Re`, 83
- `\ref`, 28, 49, 61, 70
- `remark`, 93
- `\renewcommand`, 110
- `\renewenvironment`, 110
- `report`, 22–24, 29, 43, 103, 104
- `\rho`, 80
- `\right`, 84
- `\rmfamily`, 111
- `\rVert`, 81
- `\rvert`, 80

- \scriptscriptstyle, 91
- \scriptsize, 112, 113
- \scriptstyle, 91
- \scshape, 111
- \sec, 84
- \section, 43
- \setminus, 81
- setspace, 43
- \sffamily, 111
- showidx, 107
- showlabels, 28
- \sigma, 80
- Sillabazione, 33–36, 39, 40
- \sin, 83, 84
- \singleline, 43
- \sinh, 84
- \slshape, 111
- \small, 112, 113
- smallmatrix, 87
- \smallskip, 38
- split, 87, 88
- \sqrt, 77
- subequations, 90
- subfig, 28, 73
- subfigure, 73
- \subfloat, 73
- \subparagraph, 43
- \subsection, 43
- \substack, 85
- \subsubsection, 43
- \sum, 78
- \sup, 84
- syntonly, 31
- T1, 36, 115
- table, 60–62, 66, 68–70, 72
- table*, 68
- \tableofcontents, 22, 70
- tabular, 57, 58, 60–62, 66, 86
- tabularx, 28, 59, 60
- \tan, 84
- \tanh, 84
- \TeX, 46
- \text, 76, 89
- \textasciicircum, 19
- \textasciitilde, 19
- \textbackslash, 19
- \textbf, 109, 111
- \textheight, 65
- \textit, 109, 111
- \textrm, 111
- \textsc, 111
- \textsf, 111
- \textsl, 111
- \textstyle, 91
- \texttt, 111
- \textup, 91
- \textwidth, 42, 60, 68
- \tfrac, 92
- \thanks, 48, 50
- thebibliography, 103, 104
- theindex, 107
- \theoremstyle, 93
- \theta, 80
- \tiny, 112, 113
- \title, 48
- titlepage, 23, 49
- \to, 81
- \today, 20
- \toprule, 58
- toptesi, 22
- \ttfamily, 111
- twocolumn, 23
- twoside, 23
- ucs, 35
- \underbrace, 85
- \underline, 80
- \underset, 80
- url, 28, 46
- \usepackage, 30, 111
- utf8, 35
- utf8x, 35
- \varepsilon, 80
- varioref, 28, 49
- \varphi, 80
- \varrho, 80
- \varsigma, 79, 80
- \vartheta, 80
- \vdots, 87
- \vec, 80, 86
- \verb, 55
- verbatim, 54, 55
- verse, 54

<code>\Vert</code> , 81	<code>\widehat</code> , 81
<code>\vert</code> , 80	<code>\widetilde</code> , 81
<code>\Vmatrix</code> , 86	<code>width</code> , 65
<code>\vmatrix</code> , 86	<code>xcolor</code> , 28
<code>\vref</code> , 49, 61, 70	<code>\xleftarrow</code> , 81
<code>\vspace</code> , 38	<code>\xrightarrow</code> , 81
<code>\vspace*</code> , 38	<code>Xy-pic</code> , 95