

# Una mica tanto breve introduzione a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

---

*L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> in 135 minuti*

di Tobias Oetiker

Hubert Partl, Irene Hyna e Elisabeth Schlegl

Versione 6.2, 28 febbraio 2018

Libera traduzione italiana della versione 6.2 del 30 novembre 2020

# Pagina legale aggiornata

Copyright ©1995-2018 Tobias Oetiker and Contributors. All rights reserved.

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

This document is distributed in the hope that it will be useful, but *without any warranty*; without even the implied warranty of *merchantability* or *fitness for a particular purpose*. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, see <https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html>.

Copyright ©1995-2018 Tobias Oetiker e coautori. Tutti i diritti riservati.

Questo documento è libero; è lecito ridistribuirlo e/o modificarlo secondo i termini della Licenza Pubblica Generica GNU come pubblicata dalla Free Software Foundation; o la versione 2 della licenza o (a scelta) una versione successiva.

Questo documento è distribuito nella speranza che sia utile, ma *senza* alcuna garanzia; senza neppure la garanzia implicita di commerciabilità o di applicabilità per un particolare scopo. Si veda la Licenza Pubblica Generica GNU per avere maggiori dettagli.

Ognuno dovrebbe avere ricevuto una copia della Licenza Pubblica Generica GNU insieme a questo documento; in caso contrario, la si può scaricare da <https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html>.

# Grazie!

Buona parte del materiale di questa introduzione viene da un testo introduttivo austriaco che si riferiva a L<sup>A</sup>T<sub>E</sub>X 2.09 scritto in tedesco da:

Hubert Partl <[partl@mail.boku.ac.at](mailto:partl@mail.boku.ac.at)>

*Zentraler Informatikdienst der Universität für Bodenkultur, Wien*

Irene Hyna <[Irene.Hyna@bmwf.ac.at](mailto:Irene.Hyna@bmwf.ac.at)>

*Bundesministerium für Wissenschaft und Forschung, Wien*

Elisabeth Schlegl <[noemail](mailto:noemail)>

*in Graz*

Se si desidera consultare il documento in tedesco, se ne può trovare una versione aggiornata da Jörg Knappen in [CTAN://info/lshort/german](http://CTAN://info/lshort/german); esso è distribuito con T<sub>E</sub>X Live e si trova nella cartella [\\$HOME/texlive/<year>/texmf-dist/doc/latex/lshort/lshort-german](http://$HOME/texlive/<year>/texmf-dist/doc/latex/lshort/lshort-german)

L'elenco in calce contiene i nomi delle persone che mi hanno aiutato con correzioni, suggerimenti e altro materiale per migliorare questa guida. Hanno svolto un grande lavoro collaborando a far sì che il documento raggiungesse questo stadio. Voglio sinceramente ringraziarli tutti. Naturalmente, gli errori che vi si trovano ancora sono miei. Se si trova una parola ortograficamente corretta, deve essere merito di una delle persone che me l'hanno segnalato.

Se volete contribuire a questo libretto, potete trovare tutto il codice sorgente in <https://github.com/oetiker/lshort>; i vostri *pull request* sono apprezzati. Il codice sorgente si trova anche in `CTAN://info/lshort/english` ed è distribuito con T<sub>E</sub>X Live nella cartella `$HOME/texlive/<year>/texmf-dist/doc/latex/lshort/lsshort-english/lshort-6.2.src.tar.gz`.

---

---

Eric Abrahamsen, Lenimar Nunes de Andrade, Eilinger August, Rosemary Bailey, Barbara Beeton, Marc Bevand, Connor Blakey, Salvatore Bonaccorso, Pietro Braione, Friedemann Brauer, Markus Brühwiler, Jan Busa, David Carlisle, Neil Carter, Carl Cerecke, Mike Chapman, Pierre Chardaire, Xingyou Chen, Christopher Chin, Diego Clavadetscher, Wim van Dam, Benjamin Deschwenden Jan Dittberner, Michael John Downes, Matthias Dreier, David Dureisseix, Hans Ehrbar, Elliot, Rockrush Engch, William Faulk, Robin Fairbairns, Johan Falk, Jörg Fischer, Frank Fischli, Daniel Flipo, Frank, Mic Milic Frederickx, David Frey, Erik Frisk, Hans Fugal, Robert Funnell, Greg Gamble, Andy Goth, Cyril Goutte, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Neil Hammond, Christoph Hamburger, Rasmus Borup Hansen, Joseph Hilferty, Daniel Hirsbrunner, Martien Hulsen, Björn Hvittfeldt, Morten Høgholm, Werner Icking, Eric Jacoboni, Jakob, Alan Jeffrey, Martin Jenkins, Byron Jones, David Jones, Johannes-Maria Kaltenbach, Nils Kanning, Andrzej Kawalec, Christian Kern, Alain Kessi, Axel Kielhorn, Sander de Kievit, Kjetil Kjernsmo, Tobias Klausner, Jörg Knappen, Michael Koundouros, Matt Kraai, Tobias Krewer, Flori Lambrechts, Mike Lee, Maik Lehardt, Rémi Letot, Axel Liljenkrantz, Jasper Loy, Johan Lundberg, Martin Maechler, Alexander Mai, Claus Malten, Kevin Van Maren, Pablo Markin, I. J. Vera Marín, Hendrik Maryns, Chris McCormack, Aleksandar S. Milosevic, Henrik Mitsch, Stefan M. Moser, Armin Müller, Philipp Nagele, Richard Nagy, Manuel Oetiker, Urs Oswald, Hubert Partl, Marcelo Pasin, Martin Pfister, Lan Thuy Pham, Breno Pietracci, Demerson Andre Polli, Maksym Polyakov, Nikos Pothitos, John Reffling, Mike Ressler, Brian Ripley, Kurt Rosenfeld, Bernd Rosenlecher, Chris Rowley, Young U. Ryu, Risto Saaremaa, András Salamon, José Carlos Santos, Christopher Sawtell, Gilles Schintgen, Craig Schlenter, Hanspeter Schmid, Baron Schwartz, Jordi Serra i Solanich, Miles Spielberg, Susan Stewart, Matthieu Stigler, Geoffrey Swindale, Laszlo Szathmary, Boris Tobotras, Josef Tkadlec, Scott Veirs, Didier Verna, Carl-Gustav Werner, Fabian Wernli, Matthew Widmann, David Woodhouse, Chris York, Rick Zaccone, Fritz Zaucker, and Mikhail Zotov.

La prima versione italiana, di circa vent'anni fa, è stata curata dal gruppo italiano denominato Gilda; vi contribuirono le seguenti persone:

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

La presente traduzione italiana (del 2020) è stata curata da:

cap. 1: Gianluca Pignalberi; cap. 2: Tommaso Gordini; cap. 3, 4, 5: Claudio Beccari; cap. 6: Mauro Sacchetto; revisione generale: Gianpaolo Ruocco.

# Prefazione

$\text{\LaTeX}$  [1] è un sistema di composizione tipografica molto adatto per produrre documenti scientifici e matematici di alta qualità. Esso è adatto anche per produrre ogni altro tipo di documenti, da semplici lettere a libri completi.  $\text{\LaTeX}$  usa  $\text{\TeX}$  [2] come motore di composizione.

Questa introduzione descrive  $\text{\LaTeX} 2_{\epsilon}$  e dovrebbe essere sufficiente per la maggior parte delle applicazioni di  $\text{\LaTeX}$ . Ci si riferisca ai testi [1, 3] per una sua descrizione completa.

Questo documento è diviso in sei capitoli:

**Capitolo 1** descrive la struttura di base dei documenti  $\text{\LaTeX}$ . Si esporrà anche una sua breve storia. Dopo la lettura di questo capitolo si dovrebbe avere una discreta idea di come lavora il programma.

**Capitolo 2** va nei dettagli della composizione dei documenti. Si spiega la maggior parte dei comandi e degli ambienti  $\text{\LaTeX}$  essenziali. Dopo la lettura di questo capitolo, si dovrebbe essere in grado di creare i primi documenti, contenenti elenchi puntati, tabelle, disegni e oggetti mobili.

**Capitolo 3** spiega come si compongono le formule con  $\text{\LaTeX}$ . Molti esempi mostrano come si usano i punti di forza di  $\text{\LaTeX}$ . Alla fine del capitolo ci sono le tabelle che mostrano tutti i simboli disponibili con  $\text{\LaTeX}$ .

**Capitolo 4** spiega gli indici analitici e le bibliografie e alcuni punti delicati relativi ai file PDF.

**Capitolo 5** mostra come usare  $\text{\LaTeX}$  per creare oggetti grafici. Invece di disegnare usando qualche programma esterno, salvando il file e importandolo poi nel documento, basta descrivere il disegno e lasciare che interpreti interpreti le istruzioni e le realizzi.

**Capitolo 6** contiene delle informazioni potenzialmente pericolose in merito a come modificare il layout dei documenti standard prodotti da  $\text{\LaTeX}$ . Si racconterà come cambiare le cose in modo che il bel risultato prodotto da  $\text{\LaTeX}$  diventi orrendo o straordinario, a seconda delle abilità di chi opera le modifiche.

È importante leggere i capitoli nel loro ordine – dopo tutto il libro non è così lungo. Si faccia attenzione agli esempi, perché molta dell’informazione del libro è racchiusa in essi.

L<sup>A</sup>T<sub>E</sub>X è disponibile per la maggior parte dei calcolatori, dai PC e i Mac, fino alle installazioni UNIX. Esistono anche dei siti in rete, per esempio Overleaf (<https://www.overleaf.com/>), a cui è possibile collegarsi per eseguire i programmi del sistema T<sub>E</sub>X. Sebbene sia descritto nell’appendice A, lo scopo di questo libretto non è quello di dire come installare L<sup>A</sup>T<sub>E</sub>X, bensì quello di insegnare a produrre i documenti elaborandoli con L<sup>A</sup>T<sub>E</sub>X.

Se si ha bisogno di materiale relativo al programma, si dia un’occhiata ai siti CTAN (*Comprehensive T<sub>E</sub>X Archive Network*, “Rete Completa degli Archivi di T<sub>E</sub>X”). La pagina principale si trova in <http://www.ctan.org>.

Si troveranno molti riferimenti a CTAN in tutto il libretto, specialmente link a software e documenti che si vorrebbero scaricare. Invece di scrivere l’indirizzo completo, si è scritto semplicemente CTAN seguito dalla posizione di qualunque di questi archivi si voglia connettere.

Se si vuole eseguire L<sup>A</sup>T<sub>E</sub>X sul proprio calcolatore, si veda l’appendice A. Ma, attenzione: questa succinta appendice serve solo per verificare che si sia installato il sistema T<sub>E</sub>X in modo corretto, non è un manuale di installazione.

Se avete qualche suggerimento su che cosa aggiungere, togliere o modificare in questo libretto, vi prego di farmelo sapere. Sono molto interessato specialmente a ricevere feedback dai nuovi utenti di L<sup>A</sup>T<sub>E</sub>X circa le parti di questo libretto facili da capire e quelle che potrebbero essere spiegate meglio.

Tobias Oetiker <[tobi@oetiker.ch](mailto:tobi@oetiker.ch)>

OETIKER+PARTNER AG  
Aarweg 15  
4600 Olten, Switzerland

La versione corrente di questo documento è disponibile in  
[CTAN://info/lshort](https://www.ctan.org/info/lshort).

---

Questa è una libera traduzione dell’originale di Tobias Ötiker. Può essere colpa del nostro team di traduttori se qualche cosa non è spiegata bene; ci si rivolga quindi al forum degli utenti italiani di T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X G<sub>J</sub>T: <https://www.guitex.org>.

Inoltre, la versione di Tobias Ötiker era piuttosto datata; questo team di traduttori ha cercato di aggiornare i punti più modificati rispetto alla sua versione di molti anni fa; possiamo aver commesso errori e accettiamo volentieri suggerimenti e correzioni; si prega di segnalarli nel forum del G<sub>J</sub>T.

Abbiamo aggiunto nella parte iniziale anche un glossario per spiegare certi termini del linguaggio tipografico di solito poco noti ai nuovi utenti.

# Indice

<b>Grazie!</b>	<b>iii</b>
<b>Prefazione</b>	<b>vi</b>
<b>Glossario</b>	<b>xv</b>
<b>1 Cose che bisogna sapere</b>	<b>1</b>
1.1 Un po' di storia . . . . .	1
1.1.1 T <sub>E</sub> X . . . . .	1
1.1.2 L <sup>A</sup> T <sub>E</sub> X . . . . .	2
1.2 Fondamenti . . . . .	2
1.2.1 Autore, grafico e compositore . . . . .	2
1.2.2 Progetto dell'impaginazione . . . . .	3
1.2.3 Vantaggi e svantaggi . . . . .	3
1.3 I file di input di L <sup>A</sup> T <sub>E</sub> X . . . . .	4
1.3.1 Spazi . . . . .	4
1.3.2 Caratteri speciali . . . . .	5
1.3.3 I comandi di L <sup>A</sup> T <sub>E</sub> X . . . . .	5
1.3.4 Commenti . . . . .	6
1.4 Struttura degli input . . . . .	7
1.5 Una tipica sessione della riga di comando . . . . .	9
1.6 La struttura del documento . . . . .	9
1.6.1 Classi di documento . . . . .	9
1.6.2 Pacchetti . . . . .	10
1.6.3 Stili della pagina . . . . .	12
1.7 Altri file . . . . .	12
1.8 Grandi progetti . . . . .	15
<b>2 Comporre il testo</b>	<b>17</b>
2.1 La struttura del testo e della lingua . . . . .	17
2.2 Interruzione di riga e di pagina . . . . .	19
2.2.1 Capoversi giustificati . . . . .	19
2.2.2 Sillabazione e cesura . . . . .	21
2.3 Stringhe preconfezionate . . . . .	22

2.4	Caratteri speciali e simboli	22
2.4.1	Virgolette	22
2.4.2	Tratti, trattini e lineette	23
2.4.3	Tilde ( $\sim$ )	23
2.4.4	Barra obliqua (/)	23
2.4.5	Simbolo di grado ( $^\circ$ )	24
2.4.6	Il simbolo dell'euro ( $\text{€}$ )	24
2.4.7	Ellissi (...)	25
2.4.8	Legature	25
2.4.9	Accenti e caratteri speciali	26
2.5	Supporto linguistico internazionale	26
2.5.1	Uso di <code>babel</code>	28
2.5.2	Uso di <code>polyglossia</code>	28
2.6	Lo spazio tra le parole	30
2.7	Titoli, capitoli e paragrafi	31
2.8	Riferimenti incrociati	33
2.9	Note a piè di pagina	33
2.10	Parole evidenziate	34
2.11	Ambienti	34
2.11.1	Liste: <code>itemize</code> , <code>enumerate</code> e <code>description</code>	35
2.11.2	Testi non giustificati: <code>flushleft</code> , <code>flushright</code> e <code>center</code>	35
2.11.3	Testi in evidenza: <code>quote</code> , <code>quotation</code> e <code>verse</code>	36
2.11.4	L'estratto	37
2.11.5	Stampare alla lettera	37
2.11.6	Tabelle	38
2.12	Includere figure e immagini	42
2.13	Oggetti flottanti	43
<b>3</b>	<b>Comporre matematica</b>	<b>47</b>
3.1	Il pacchetto <code>amsmath</code>	47
3.2	Equazioni isolate	47
3.2.1	Il modo matematico	50
3.3	Come si costruisce una formula matematica	51
3.4	Singole equazioni troppo lunghe: <code>multline</code>	56
3.5	Insiemi di equazioni	57
3.5.1	Problemi con i comandi tradizionali	58
3.5.2	L'ambiente <code>align</code>	60
3.6	Matrici	61
3.7	Le spaziature in modo matematico	63
3.8	I caratteri fantasma	65
3.9	Giocare con i font matematici	65
3.9.1	I simboli neri	66
3.10	Teoremi, lemmi, e altri enunciati	66
3.11	Dimostrazioni e il simbolo di "fine dimostrazione"	68

3.12	Elenchi di simboli matematici . . . . .	70
<b>4</b>	<b>Funzionalità speciali</b>	<b>79</b>
4.1	Bibliografia . . . . .	79
4.2	Indici analitici . . . . .	80
4.3	Testatine e piedini personalizzati . . . . .	82
4.4	Il pacchetto <code>verbatim</code> . . . . .	83
4.5	Installare altri pacchetti . . . . .	84
4.6	$\LaTeX$ e PDF . . . . .	85
4.6.1	Collegamenti ipertestuali o <i>hyperlink</i> . . . . .	86
4.6.2	Problemi con i link . . . . .	89
4.6.3	Problemi con i segnalibri . . . . .	89
4.7	Lavorare con $X_{\text{T}}\LaTeX$ e $\text{Lua}\LaTeX$ . . . . .	90
4.7.1	I font . . . . .	90
4.7.2	Dove si trovano i font OpenType? . . . . .	92
4.7.3	Compatibilità fra $X_{\text{T}}\LaTeX$ o $\text{Lua}\LaTeX$ e $\text{pdf}\LaTeX$ . . . . .	92
4.8	Le presentazioni . . . . .	93
<b>5</b>	<b>Creare la grafica per la matematica</b>	<b>97</b>
5.1	Osservazioni introduttive . . . . .	97
5.2	L'ambiente <code>picture</code> . . . . .	98
5.2.1	Comandi di base . . . . .	99
5.2.2	I segmenti . . . . .	100
5.2.3	Frecce . . . . .	103
5.2.4	Cerchi e circonferenze . . . . .	104
5.2.5	Spessori delle linee . . . . .	105
5.2.6	Fine dei percorsi e loro giunzioni . . . . .	105
5.2.7	Testo e formule . . . . .	107
5.2.8	<code>\multiput</code> e <code>\linethickness</code> . . . . .	108
5.2.9	Rettangoli arrotondati . . . . .	108
5.2.10	Uso ripetitivo di figure parziali . . . . .	109
5.2.11	Curve di Bézier quadratiche . . . . .	110
5.2.12	Catenaria . . . . .	111
5.2.13	La rapidità nella teoria della relatività speciale . . . . .	112
5.3	Le collezioni PGF e <code>TikZ</code> . . . . .	113
<b>6</b>	<b>Personalizzare <math>\LaTeX</math></b>	<b>117</b>
6.1	Comandi, ambienti e pacchetti nuovi . . . . .	117
6.1.1	Nuovi comandi . . . . .	118
6.1.2	Nuovi ambienti . . . . .	119
6.1.3	Spazio extra . . . . .	120
6.1.4	Pacchetti personalizzati . . . . .	120
6.2	Caratteri e corpi . . . . .	121
6.2.1	Comandi per cambiare carattere . . . . .	121

---

6.2.2	Pericolo, pericolo, ...	124
6.2.3	Consiglio	125
6.3	Spaziature	125
6.3.1	Scartamento o avanzamento di riga	125
6.3.2	Formattazione dei capoversi	126
6.3.3	Spazio orizzontale	127
6.3.4	Spazio verticale	128
6.4	Impostazione della pagina	129
6.5	Altro divertimento con le lunghezze	131
6.6	Scatole	132
6.7	Linee	135
<b>A</b>	<b>Installazione di L<sup>A</sup>T<sub>E</sub>X</b>	<b>137</b>
A.1	Che cosa installare	137
A.2	Editor multiplatforma	137
A.3	T <sub>E</sub> X su Mac	138
A.4	T <sub>E</sub> X su Windows	138
A.5	T <sub>E</sub> X su Linux	139
A.6	Raccomandazioni	140
A.7	L'editor	140
A.8	Lavorare con i file grafici	140
	<b>Bibliografia</b>	<b>143</b>
	<b>Riferimenti italiani</b>	<b>146</b>
	<b>Indice analitico</b>	<b>148</b>

# Elenco delle figure

1.1	Un file $\LaTeX$ minimo . . . . .	8
1.2	Esempio di un articolo di rivista realistico . . . . .	8
2.1	La gestione dei font e delle lingue . . . . .	28
4.1	Esempio di impostazione di <code>fancyhdr</code> . . . . .	83
4.2	Legature antiche . . . . .	91
4.3	Due quadri con sei schermate . . . . .	95
6.1	Parametri per l'impostazione delle pagine di questo libro . . .	130

# Elenco delle tabelle

1.1	Classi di documento . . . . .	10
1.2	Opzioni per la classe di documento . . . . .	11
1.3	Alcuni dei pacchetti distribuiti con $\text{\LaTeX}$ . . . . .	13
1.4	Gli stili di pagina predefiniti di $\text{\LaTeX}$ . . . . .	14
2.1	Una borsa piena di simboli dell'euro . . . . .	25
2.2	Accenti e caratteri speciali . . . . .	27
2.3	Alcune chiavi per il pacchetto <code>graphicx</code> . . . . .	43
2.4	Autorizzazioni di posizionamento . . . . .	44
3.1	Comandi per i nomi di alcune funzioni matematiche . . . . .	53
3.2	Accenti matematici . . . . .	71
3.3	Lettere greche . . . . .	71
3.4	Relazioni binarie . . . . .	72
3.5	Operatori binari . . . . .	72
3.6	Grandi operatori . . . . .	73
3.7	Frecce . . . . .	73
3.8	Grandi frecce e accenti . . . . .	73
3.9	Delimitatori . . . . .	74
3.10	Per comporre alcuni grandi delimitatori . . . . .	74
3.11	Altri simboli . . . . .	74
3.12	Simboli non matematici . . . . .	74
3.13	$\mathcal{AMS}$ : delimitatori accessibili con il pacchetto <code>amssymb</code> . . . . .	75
3.14	$\mathcal{AMS}$ : lettere greche ed ebraiche accessibili con il pacchetto <code>amssymb</code> . . . . .	75
3.15	Alfabeti matematici . . . . .	75
3.16	$\mathcal{AMS}$ : operatori binari accessibili con il pacchetto <code>amssymb</code> . . . . .	75
3.17	$\mathcal{AMS}$ : Relazioni binarie accessibili con il pacchetto <code>amssymb</code> . . . . .	76
3.18	$\mathcal{AMS}$ : frecce accessibili con il pacchetto <code>amssymb</code> . . . . .	77
3.19	$\mathcal{AMS}$ : frecce e relazioni binarie negate accessibili con il pacchetto <code>amssymb</code> . . . . .	78
3.20	$\mathcal{AMS}$ : altri simboli accessibili con il pacchetto <code>amssymb</code> . . . . .	78
4.1	Indice analitico: sintassi e alcuni esempi . . . . .	81

6.1	Caratteri . . . . .	122
6.2	Dimensioni . . . . .	122
6.3	Corpi in punti . . . . .	123
6.4	Caratteri matematici . . . . .	123
6.5	TEX Unità di misura . . . . .	128

# Glossario

Nella composizione tipografica si usano molti termini tecnici; è bene conoscerne il significato.

$\LaTeX$ , come è noto, è una specie di linguaggio di programmazione che serve per descrivere i contenuti di un testo da comporre tipograficamente. Questo avviene mediante istruzioni che portano nomi diversi; avviene mediante vari “ambienti”, e questa parola di per sé non ha il suo significato letterale; eccetera.

Il linguaggio di programmazione  $\LaTeX$  è compreso da un programma che lo interpreta e che dispone il materiale da comporre secondo altre istruzioni che si trovano già predisposte nei file di classe e in diversi pacchetti, talvolta costituiti da un solo file, talvolta da centinaia. Perciò bisogna distinguere bene la terminologia in modo da sapere con precisione di che cosa si sta parlando.

Alcuni termini sono sinonimi di altri; per conoscerne il significato basta leggere la spiegazione del sinonimo “capostipite”.

Una rapida scorsa a questo glossario può essere utile fin dall’inizio; ma lo si usi come risorsa per cercare, via via che le si incontra, il significato di parole o locuzioni che non si conoscono bene o non si conoscono affatto.

In tutto il testo e anche in questo glossario l’indicazione  $\langle parola \rangle$  indica un argomento che deve essere sostituito dall’utente (comprese le parentesi unciniate) con qualcosa di pertinente al suo documento.

**ambiente** Codice sorgente compreso dentro un gruppo delimitato dalle istruzioni  $\backslash\text{begin}\{\langle ambiente \rangle\}$  e  $\backslash\text{end}\{\langle ambiente \rangle\}$ . Per la matematica gli ambienti testuali possono avere il medesimo aspetto, ma ce ne sono altri specifici come  $\mathcal{\langle matematica \rangle}$ , oppure come  $\llbracket \langle matematica \rangle \rrbracket$ .

**ascendente** Parte di un carattere che sporge sopra la linea orizzontale appoggiata sulla lettera ‘x’.

**avanzamento di riga** Sinonimo di scartamento.

**bounding box** È una locuzione standardizzata che in italiano si potrebbe tradurre con *rettangolo circoscritto*; è il rettangolo ideale che racchiude una figura, una tabella, una parola, un intero capoverso. Le sue dimensioni sono spesso specificate con le coordinate del vertice inferiore

sinistro e di quelle del vertice superiore destro, in sostanza le coordinate degli estremi della diagonale SO–NE del rettangolo.

**carattere** Designa un singolo glifo di un determinato font; talvolta (al plurale) viene usato per indicare una collezione di caratteri. Ogni carattere ha una larghezza sua specifica; i caratteri della stringa ‘bdfhilt’ presentano degli *ascendenti* e risultano più alti delle lettere minuscole presenti nella stringa ‘acemnorsuvwxz’; tutti questi caratteri sono appoggiati idealmente su una linea orizzontale che si chiama *linea di base*; invece i caratteri della stringa ‘gpqyQ’ presentano dei *discendenti* che sporgono sotto la linea di base.

**carattere attivo** I caratteri specificati come attivi si comportano come delle istruzioni e possono essere definiti o ridefiniti a piacere. L’unico carattere attivo presente nel nucleo di L<sup>A</sup>T<sub>E</sub>X è la tilde ~ che corrisponde all’istruzione di inserire uno spazio interparola insecabile. Molti pacchetti definiscono altri caratteri attivi; quasi tutti i file di descrizione delle lingue, compreso il file che descrive l’italiano, definiscono attivo il doppio apice diritto " .

**carattere di controllo** Un carattere di controllo è un’istruzione formata dal segno \ seguito da un carattere alfabetico, per esempio “\, ”.

**codifica** Regola con la quale i glifi sono associati a codici numerici perché i programmi dei calcolatori (numerici) possano riferirsi al segno giusto da inserire in un testo.

**codifica dei font** Codifica usata per identificare un carattere all’interno di un file che contiene i loro “disegni”. Lavorando con pdfL<sup>A</sup>T<sub>E</sub>X (vedi), la codifica di ingresso è diversa dalla codifica dei font con cui si compone il testo. Le codifiche dei font sono molto diverse l’una dall’altra. Opportuni pacchetti, *inputenc* e *fontenc*, permettono di trasferire i caratteri con la codifica di entrata nella codifica dei caratteri dei font usati per la composizione. Oggi, lavorando con X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X e LuaL<sup>A</sup>T<sub>E</sub>X, si usano i font OpenType che sono codificati nello stesso modo della codifica d’entrata UTF-8.<sup>1</sup>

**codifica di ingresso** Codifica usata dallo *shell editor* per immettere i caratteri digitati sulla tastiera nel file sorgente di un documento da comporre con i programmi del sistema T<sub>E</sub>X. Le codifiche di entrata usabili con i programmi di composizione del sistema T<sub>E</sub>X sono numerose e, tranne una che è del tutto generale e non dipende dalla marca del calcolatore né dal sistema operativo, sono tutte diverse l’una dall’altra; per motivi

---

<sup>1</sup>Più precisamente, UTF-8 è una transcodifica di quella nota col nome Unicode; la codifica UTF-8 è più semplice della Unicode e meno ingombrante in memoria, ma ai fini dell’utente finale di solito si possono considerare equivalenti.

storici, oltre alle codifiche più recenti sono ancora disponibili codifiche che venivano usate con i sistemi operativi di una volta. Tra le tante, l'unica da preferire *sempre* con pdfL<sup>A</sup>T<sub>E</sub>X quando si compongono documenti nuovi è la codifica UTF-8; la si specifica al pacchetto `inputenc` mediante l'opzione `utf8`. Lavorando con X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X e LuaL<sup>A</sup>T<sub>E</sub>X essa è l'unica codifica consentita.

**comando nativo** Sono comandi nativi tutte le istruzioni facenti parte del programma che esegue l'interpretazione del codice dell'utente e lo trasforma in codice macchina eseguendolo direttamente; per esempio `\hbox` è un comando nativo dei programmi-interpreti `tex`, `etex`, `pdftex`, `xetex`, `luatex`. Invece, `\mbox` è un comando definito da L<sup>A</sup>T<sub>E</sub>X che prima di produrre l'effetto, deve venire interpretato e trasformato in una sequenza di comandi nativi.

**comando primitivo** Sinonimo di “comando nativo”.

**controllo, stringa di** La stragrande maggioranza delle macro e dei comandi nativi sono formati da una stringa *puramente alfabetica* preceduta dal segno `\`; per esempio, la stringa `chapter` produce la macro `\chapter`. Il concetto di “stringa di controllo” non si applica alle macro definite con i caratteri di controllo.

**corpo dei caratteri** Il corpo dei caratteri indica la loro altezza complessiva, ma era un concetto preciso quando si usavano i caratteri metallici, che erano allineabili proprio perché i blocchetti di metallo avevano tutti la faccia portacarattere della stessa altezza. Con i font elettronici il concetto di corpo è più evanescente, ma in linea di massima si potrebbe pensare che il corpo sia l'altezza complessiva dei glifi testuali più grandi, di solito le parentesi tonde. Per i font matematici il concetto di corpo è spesso solo convenzionale.

**corpo del documento** Il corpo di un documento si chiama anche “parte principale” e si trova fra la parte iniziale e quella finale del documento.

**corpo del testo** Il corpo del testo in una pagina è la parte scritta sulla maggior parte del foglio fra la testatina e il piedino.

**discendente** Parte di un carattere che sporge sotto la linea di base.

**file di classe** Si tratta di un file che descrive in senso generale una classe di documenti; un particolare documento può essere ulteriormente specificato mediante le *opzioni della classe* o mediante l'uso di “pacchetti” di estensione. Esiste la classe `book` per comporre i libri che hanno una struttura intrinseca piuttosto complessa, e una classe `article` che

descrive brevi documenti con una struttura intrinseca molto più semplice. Il sistema  $\text{\TeX}$  fornisce numerosi file di classe con caratteristiche generali molto diverse. I file di classe hanno estensione `.cls`.

**file di definizioni** Si tratta di raccolte di definizioni di vario genere. Questi file spesso hanno estensione `.def`.

**file di definizioni dei font** Sono file con estensione `.fd` che contengono le specifiche per usare font con certe codifiche, appartenenti a certe famiglie di font, con serie di “nero” e con disegni diversi ma dello stesso stile, e con corpi diversi.

**file di descrizione delle lingue** Sono file con estensione `.ldf` che contengono le parole fisse di una data lingua (come “Capitolo”, “Indice”, “Appendice”, eccetera) e le impostazioni tipografiche usate nelle nazioni dove si stampa prevalentemente in quella lingua.

**file di stile** Sinonimo di “pacchetto”. Sebbene l’estensione `.sty` caratterizzi i pacchetti, questa scelta è una questione di retrocompatibilità per continuare a usare i file prodotti per il vecchio  $\text{\LaTeX}$  2.09, abbandonato nel 1994 perché sostituito con  $\text{\LaTeX}$  2 $\epsilon$ . In un futuro che si spera prossimo,  $\text{\LaTeX}$  2 $\epsilon$  verrà sostituito da  $\text{\LaTeX}$  3, di cui esistono già diversi moduli a cui molte classi e molti pacchetti si appoggiano.

**facciata** Un lato di un foglio stampato; può essere il recto o il verso del foglio.

**font** È un file con estensioni di vario genere a seconda di come sono rappresentati i disegni dei caratteri che descrive; esso contiene anche le informazioni metriche sulle dimensioni di ogni segno o glifo, e sulle proprietà che ogni glifo ha in relazione alla sua posizione in vicinanza di certi altri glifi; si tratta delle crenature o accostamenti e delle legature; per esempio, V e A non vengono composti come VA, ma come VA, avvicinando la A alla V, per non lasciare un inestetico spazio troppo grande fra le due lettere; per esempio una ‘f’ seguita da un’altra ‘f’, non produce ‘ff’ ma ‘ff’, con le due lettere legate in un unico glifo. Oggi con i font elettronici non si usa più la locuzione *polizza di caratteri*, che veniva usata ai tempi dei caratteri metallici. In un certo senso le due locuzioni si assomigliano, ma non indicano esattamente la stessa cosa, perché la polizza non conteneva solo la collezione dei segni stampabili, ma anche un numero preciso di ciascun esemplare dello stesso segno. Con i font elettronici non c’è limite al numero di segni identici che si possono usare.

**font bitmapped** I font *bitmapped*, o *emphraster*, o a matrici di punti sono fatti mediante singoli puntini (di solito di forma rettangolare) allineati

lungo una scacchiera ideale, in modo che la visione dell'intera matrice dia l'immagine del glifo. Questa è perfetta se la densità delle maglie della scacchiera è molto fine e adatta al supporto su cui ogni glifo deve apparire in relazione al dispositivo che lo rende visibile. Questi font bitmapped non possono essere ingranditi e rimpiccioliti a piacere perché continuano ad essere rappresentati dagli stessi rettangolini riportati nella stessa scacchiera con le maglie ingrandite o rimpicciolite. Ingrandendo il glifo, se ne vedono chiaramente i bordi obliqui pesantemente seghettati.

**font graziato** Indica un font dotato di grazie, cioè di quelle decorazioni alle estremità dei disegni dei glifi che permettono di vedere meglio le singole lettere e di distinguere meglio lo spazio fra le parole.

**font lineare** Font privo di grazie come in font lineare. Il font lineare prende anche altri nomi, come font bastoncino, font sans serif, font sgraziato.

**font monospaziato** Sinonimo di font *teletype*.

**font proporzionale** I font proporzionali sono quelli che usano spazi diversi in proporzione alla larghezza di ogni glifo. Nel font usato in questo capoverso la lettera 'M' usa uno spazio ben maggiore di quello della lettera 'i'.

**font teletype** Le macchine da scrivere di un tempo e le telescriventi anche moderne scrivono con font battuti con martelletti che imprimono il loro disegno sul nastro inchiostrato, in modo che l'inchiostro si trasferisca sulla carta con lo stesso segno; per ragioni meccaniche, i martelletti hanno la stessa larghezza e il carrello portacarta si muove con passi della stessa lunghezza; le lettere impresse hanno quindi la stessa larghezza **facilmente riconoscibile**. Verso la fine della produzione delle macchine da scrivere, apparvero le macchine con i caratteri in rilievo su una testina sferica mobile, oppure in rilievo sui "petali" di una grande "margherita"; questa sfera o questa margherita si spostava lungo il carrello portacarta lasciandolo fermo, ma il dispositivo portasfera o portamargherita continuava a spostarsi a scatti di passo costante. Migliorando la tecnologia elettromeccanica, apparvero poi anche le macchine da scrivere proporzionali, ma la loro epoca durò poco per l'avvento dei PC. La telescrivente (*teletype* in inglese, da cui la sigla **tt** per questi font) sembra che sia rimasta alla versione ad avanzamento costante del carrello.

**font vettoriale** I font vettoriali sono specificati con una descrizione della forma geometrica del contorno dei loro glifi. In questo modo, cambiando semplicemente il fattore di scala delle coordinate del disegno geometrico

del contorno, ogni glifo può venire ingrandito o rimpicciolito a piacere senza perdere nulla nel suo aspetto.

**gabbia** Rettangolo ideale dentro il quale si compone il testo di una pagina.

**gabbia esterna** Rettangolo ideale che comprende tutto quanto è stampato sulla pagina, compresi testatina e piedino.

**gabbia interna** Rettangolo ideale che comprende solo il testo presente nella pagina, escludendo quindi testatina e piedino.

**geometria della pagina** Spazio e posizione delle parti stampate sulla facciata del foglio; non riguarda solo le dimensioni della gabbia interna o esterna che sia, ma comprende le dimensioni dei margini, le dimensioni relative della gabbia interna rispetto alla gabbia esterna e la posizione di quella dentro questa; la posizione delle note marginali; la posizione e distanza dal corpo del testo dalle note al piede; le dimensioni dei font nonché le loro proprietà; eccetera. Il *book design*, o grafica editoriale, è un'arte difficile; un dilettante che vi si cimenti verosimilmente all'inizio commette errori grossolani, che impara a evitare via via che osserva e valuta il disegno editoriale delle migliori case editrici, sia esaminando le pubblicazioni moderne, sia quelle dei tempi andati.

**glifo** Un glifo è il disegno specifico di un carattere; i glifi *a*, **a**, *a*, **A** sono diversi, anche se tutti fanno riferimento alla stessa lettera.

**interlinea** Termine spesso scorrettamente usato per indicare lo scartamento; l'interlinea invece è lo spazio presente *fra* due righe consecutive di testo. L'interlinea è sempre una piccola frazione del corpo dei caratteri con cui sono composte le due righe. Al contrario lo scartamento è solitamente maggiore del corpo e molto raramente ne è una frazione quasi unitaria. Se l'interlinea è positiva, il testo si dice *interlineato*; se l'interlinea è negativa (quindi lo scartamento è leggermente minore del corpo) il testo si dice *sterlineato*. Il testo sterlineato si usa solo quando si compongono titoli usando lettere maiuscole e, talvolta, quando si compone in maiuscoletto.

**istruzione** Sono istruzioni tutti i comandi nativi e le macroistruzioni definite dai file contenenti lo specifico *mark up* o definite dall'utente. Sono istruzioni, quindi, sia le stringhe di controllo, sia i caratteri di controllo, sia i caratteri attivi.

**L<sup>A</sup>T<sub>E</sub>X** È uno speciale linguaggio di programmazione, definito *mark up*; esistono molti linguaggi di *mark up* e, forse, uno dei più noti è il linguaggio HTML finalizzato alle schermate della rete; L<sup>A</sup>T<sub>E</sub>X lo è alla composizione tipografica; sono diversi i programmi che lo interpretano le sue istruzioni per trasformarle in comandi per la macchina che deve

tradurre il tutto in un file in formato PDF; in questo modulo ne abbiamo nominati diversi, ma ci siamo occupati in pratica solo di pdfL<sup>A</sup>T<sub>E</sub>X. Il vero interprete del linguaggio si chiama `pdftex` ed è configurato per conoscere il linguaggio L<sup>A</sup>T<sub>E</sub>X. Talvolta si dice L<sup>A</sup>T<sub>E</sub>X per dire pdfL<sup>A</sup>T<sub>E</sub>X; come dovrebbe essere chiaro da quanto detto, si tratta di due cose diverse e si suggerisce di tenere ben distinti i due nomi. Si ricorda che il nome di questo mark up si pronuncia /lâtek/ oppure /lâtekh/, assolutamente non /laték/ o, meno che mai, /latex/.<sup>2</sup> Oggi con L<sup>A</sup>T<sub>E</sub>X non ci si riferisce più alla versione del mark-up abbandonata nel 1994, ma a quella nuova che prende la sigla di L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. Da circa un decina d'anni si sta lavorando a L<sup>A</sup>T<sub>E</sub>X3, siglato L3; l'utente non si preoccupi di dover imparare tanti linguaggi; a livello utente, il mark-up è sempre quello siglato L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, il solo di cui si parla in questa guida.

**linea di base** Linea retta ideale su cui sono allineate le lettere minuscole prive di discendenti che formano una riga di testo.

**macro** Abbreviazione di macroistruzione; è usata come suo sinonimo.

**macroistruzione** Istruzione ottenuta per definizione da parte del file di mark-up, dai file di classe, dai file di estensione (o pacchetti) o dall'utente; è definita mediante sequenze di altre macroistruzioni più semplici e/o di comandi nativi.

**mark up** Metodo per descrivere le parti di un testo perché un calcolatore le possa mostrare sullo schermo o su carta in modo coerente e comprensibile alle persone che leggono quel testo. Il sistema T<sub>E</sub>X usa i mark up Plain, L<sup>A</sup>T<sub>E</sub>X, Aleph, Lamed, ConT<sub>E</sub>Xt e altri molto meno conosciuti.

**nome di un comando** È quanto resta dopo aver tolto la barra inversa iniziale ad un comando. Il nome del comando `\chapter` è `chapter`; è quasi un sinonimo di “sequenza di controllo”.

**occhio** Termine tecnico usato in tipografia che indica la dimensione verticale dei caratteri di uno stesso font. Non è sinonimo di corpo. Al tempo dei caratteri metallici, era l'altezza del disegno in rilievo dei segni sulla faccia dei blocchetti metallici.

**occhio mediano** Riprende il concetto di ‘occhio’, ma si riferisce all'altezza delle lettere senza ascendenti né discendenti; a pari corpo, molti font differiscono solo per l'occhio mediano. Tale differenza è direttamente visibile confrontando i caratteri tondi graziati della parola ‘accesso’ con quelli tondi sgraziati della stessa parola ‘**accesso**’: il corpo del carattere è lo stesso, 11 pt, ma il font è diverso (Helvetica nel secondo caso).

---

<sup>2</sup>L'indicazione */⟨suono⟩/* serve per rappresentare con lettere o alti simboli speciali il *⟨suono⟩* di una parola.

- pacchetto** Un pacchetto può essere costituito da un solo file con estensione `.sty` o un insieme di file subordinati a uno o più particolari file con quella estensione.
- pagina** Una pagina in senso stretto è sinonimo di foglio; tuttavia, viene usata quasi sempre nel significato di facciata.
- parte finale** La parte finale di un documento contiene la bibliografia, uno o più indici analitici e simili sezioni di consultazione.
- parte iniziale** La parte iniziale di un documento può essere strutturata in vari modi, ma di solito contiene la pagina del titolo (chiamata anche frontespizio), la pagina legale, una possibile presentazione scritta da una persona diversa dall'autore e/o una descrizione del lavoro predisposta dall'autore, l'indice generale, una possibile introduzione. Talvolta, la parte iniziale ha le pagine numerate con numeri romani.
- pdf $\LaTeX$**  Indica un programma costituito dall'interprete `pdftex` e dal codice del mark up  $\LaTeX$ . Normalmente pdf $\LaTeX$  legge una sequenza di file testuali scritti in linguaggio  $\LaTeX$  e produce in uscita un documento composto in formato PDF. Questo programma lavora con font codificati con 8 bit, tipicamente i font Adobe Type 1, ma tutti limitati a 256 caratteri. Non lavora direttamente con i font OpenType, codificati Unicode che, in teoria, potrebbero contenere milioni di glifi; con questi font bisogna usare i programmi X $\LaTeX$  e Lua $\LaTeX$ .
- piegino** Rappresenta la riga scritta sotto il corpo del testo in ogni pagina; spesso il piegino è vuoto, oppure contiene solo il numero della pagina; in certi documenti il piegino può essere molto complesso e strutturato.
- recto** Il recto è la facciata anteriore di un foglio; in un libro, la pagina di destra mostra il suo recto numerato con un numero dispari. In effetti, parlare di recto o di pagina dispari nei libri è la stessa cosa, ma la parola 'recto' è molto più breve.
- rettangolo circoscritto** sinonimo di *bounding box*.
- scartamento** Questa parola indica la distanza fra le linee di base due righe successive di testo composte con lo stesso font. Lo scartamento è (quasi) sempre maggiore o uguale al corpo del testo; nel gergo tipografico si dice che un testo è composto in corpo 10/12 intendendo che il font ha un corpo di 10 pt e che lo scartamento vale 12 pt. Quando lo scartamento è (leggermente) minore del corpo, si dice che il testo è *sterlineato*; ai tempi della composizione con i caratteri metallici, comporre in modo sterlineato era molto difficile o richiedeva font fusi apposta; oggi è facilissimo farlo con i font elettronici, e lo si fa (in realtà raramente)

---

quando si compone in tutte maiuscole o in maiuscoletto, perché questi glifi sono sostanzialmente privi di parti discendenti.

**spazio insecabile** È uno spazio fra le parole prima del quale il programma di composizione non può eseguire un fine-riga, cioè non può andare a capo.

**shell editor** È un programma per creare e modificare testi che dispone di adeguate funzionalità per agevolare l'utente nell'esecuzione dei comandi del sistema operativo o dei comandi *batch* installati sulla propria macchina. I comandi *batch* sono quelli che l'utente potrebbe far eseguire al proprio calcolatore, aprendo una finestra, o prompt comandi, o terminale, scrivendoci dentro il comando con i suoi argomenti, e premendo il tasto Invio. I programmi del sistema T<sub>E</sub>X sono quasi tutti di tipo *batch*, quindi per lavorare con i file sorgente da far loro elaborare, è utilissimo usare uno shell editor che permetta all'utente di scegliere una icona o una voce di un menù, che si occupa di trasmettere al sistema operativo i comandi e i loro argomenti, risparmiando all'utente la noia di scriverli e riscriverli e, ancora più importante, evitandogli di commettere errori di digitazione.

**specchio di stampa** È una locuzione del tutto equivalente a “griglia di stampa”.

**stringa di controllo** La maggior parte delle macro hanno un nome formato solo da lettere precedute dal segno \; la stringa letterale che rappresenta il nome della macro termina col primo segno diverso da una lettera. Per lettere si intendono solo le 52 lettere dell'alfabeto latino<sup>3</sup> (26 maiuscole e 26 minuscole) prive di qualunque segno diacritico.

**testatina** La testatina è quella particolare riga di testo in testa a ogni pagina dove possono essere scritte informazioni che rendono più facile sfogliare un documento per trovarne più rapidamente una determinata sezione. È complementare all'indice generale. In un dizionario, la testatina è essenziale per trovare i singoli lemmi.

**verso** Il verso di un foglio è la sua faccia posteriore; in un libro le pagine mostrano il recto a destra, numerato con un numero dispari; il verso è la sua facciata posteriore, marcata con un numero pari.

**x-height** Letteralmente l'altezza della lettera minuscola 'x'; nel gergo tipografico italiano prende il nome di *occhio mediano*.

---

<sup>3</sup>Per 'alfabeto latino' si intende universalmente l'alfabeto composto dalle lettere 'a b c d e f g h i j k l m n o p q r s t u v w x y z' e dalle corrispondenti maiuscole; è noto, però, che i Romani non conoscevano le lettere 'j w' e scrivevano lo stesso segno 'V' per rappresentare tre fonemi diversi.



# Capitolo 1

## Cose che bisogna sapere

La prima parte di questo capitolo presenta una breve panoramica della filosofia e della storia di  $\LaTeX 2_{\epsilon}$ . La seconda parte si focalizza sulle strutture di base di un documento  $\LaTeX$ . Dopo aver letto questo capitolo, dovrete avere una conoscenza grezza di come lavora  $\LaTeX$ , cosa che vi servirà per capire il resto del libro.

### 1.1 Un po' di storia

#### 1.1.1 $\TeX$

$\TeX$  è un programma per computer creato da Donald E. Knuth [2]. È orientato alla composizione tipografica di testo e formule matematiche. Knuth iniziò a scrivere il motore di composizione  $\TeX$  nel 1977 per esplorare il potenziale delle apparecchiature per la stampa digitale che stavano cominciando a inserirsi nell'industria editoriale a quel tempo, specialmente nella speranza di poter invertire la tendenza al deterioramento della qualità tipografica che vedeva affliggere i propri libri e articoli. Il  $\TeX$  che usiamo oggi è stato lanciato nel 1982, con alcune leggere migliorie aggiunte nel 1989 per meglio supportare i caratteri a 8 bit e diverse lingue.  $\TeX$  è rinomato per essere estremamente stabile, per girare su diversi tipi di computer e per essere virtualmente libero da bug. Il numero di versione di  $\TeX$  converge a  $\pi$  e ora è 3,141592653.

$\TeX$  si pronuncia “tech”, con un “ch” come nella parola tedesca “ach”<sup>1</sup> o come si pronuncia a Firenze la parola “casa”. Il “ch” deriva dall'alfabeto

---

<sup>1</sup>In tedesco ci sono in realtà due pronunce di “ch” e si potrebbe supporre che il suono morbido del “ch” di “Pech” potrebbe essere più appropriato. Interrogato su ciò, Knuth scrisse su Wikipedia tedesca: *Non mi arrabbio quando le persone pronunciano  $\TeX$  nel modo che preferiscono. . . e in Germania molti usano il ch morbido perché la X segue la vocale e, non il ch duro che segue la vocale a. In Russia ‘tex’ è una parola molto comune, pronunciata ‘tjekh’. Ma credo che la pronuncia più appropriata sia udibile in Grecia, dove hanno il ch aspro di ach e Loch.*

greco dove “X” è la maiuscola della lettera “χ”.  $\text{T}_{\text{E}}\text{X}$  è anche la prima sillaba della parola greca τέχνη (tecnica, arte). In un ambiente ASCII,  $\text{T}_{\text{E}}\text{X}$  diventa  $\text{TeX}$ .

### 1.1.2 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  abilita gli autori a comporre e stampare i propri lavori con la più alta qualità tipografica usando un’impaginazione professionale predefinita.  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  fu scritto in origine da Leslie Lamport [1]; usa il programma  $\text{T}_{\text{E}}\text{X}$  come motore di composizione.  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  è correntemente mantenuto dal  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  Project.

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  in italiano si pronuncia “làtek”. Se ci si riferisce a  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  in un ambiente ASCII si scrive  $\text{LaTeX}$ .  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$  si pronuncia “làtek due e” e in ASCII si scrive  $\text{LaTeX2e}$ .

## 1.2 Fondamenti

### 1.2.1 Autore, grafico e compositore

Per pubblicare qualcosa, gli autori forniscono il loro manoscritto a una casa editrice. Uno dei loro grafici decide poi l’aspetto del documento (ampiezza delle colonne, caratteri, spazio prima e dopo le testatine, ...). Il grafico scrive le sue istruzioni sul manoscritto, quindi lo passa a un compositore che compone il libro in base a quelle istruzioni.

Il grafico umano prova a capire quel che aveva in mente l’autore mentre scriveva il manoscritto. Prende decisioni circa la forma e lo stile relativi a intestazioni dei capitoli, citazioni, esempi, formule, eccetera, in base alla propria esperienza professionale e al contenuto del manoscritto.

In un ambiente  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ , l’autore usa  $\text{T}_{\text{E}}\text{X}$  come compositore. Ma  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  è “solo” un programma e quindi necessita di maggiori istruzioni. L’autore deve pertanto fornire informazioni aggiuntive per descrivere la struttura logica del suo lavoro. Queste informazioni vengono scritte nel testo come “comandi  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ”.

Ciò è abbastanza diverso dall’approccio WYSIWYG<sup>2</sup> usato dai più moderni elaboratori di testo come *MS Word* o *LibreOffice*. In questi applicativi, gli autori specificano interattivamente l’aspetto del documento mentre digitano il testo. Essi possono vedere sullo schermo l’aspetto che il lavoro finito (probabilmente, visti i risultati spesso aleatori ottenuti con questi applicativi) avrà una volta stampato.

Usando  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  non è normalmente possibile vedere il risultato finale mentre si digita il testo, ma detto risultato può essere visualizzato sullo schermo dopo aver elaborato il file con  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ . Quindi si potranno fare le correzioni prima di mandare realmente il documento in stampa.

---

<sup>2</sup>What You See Is What You Get – Quello che vedi è quello che ottieni.

### 1.2.2 Progetto dell'impaginazione

La progettazione tipografica è un mestiere. Gli autori non addestrati commettono spesso gravi errori di formattazione assumendo che la progettazione di un libro sia perlopiù una questione di estetica: “Se un documento sembra artisticamente piacevole, è ben progettato”. Ma siccome un documento dev'essere letto e non appeso in una galleria, la leggibilità e la comprensibilità sono molto più importanti del suo bellissimo aspetto. Esempi:

- La dimensione del carattere e la numerazione delle testatine (gli elementi testuali e grafici situati superiormente alle pagine) devono essere scelti in modo da rendere chiara al lettore la struttura dei capitoli e delle sezioni.
- La lunghezza delle righe dev'essere corta abbastanza da non affaticare gli occhi del lettore, ma lunga abbastanza da riempire piacevolmente la pagina.

Col sistema WYSIWYG, gli autori generano spesso documenti esteticamente piacevoli con strutture povere o inconsistenti.  $\text{\LaTeX}$  previene questi errori di formattazione costringendo l'autore a dichiarare la struttura *logica* del suo documento.  $\text{\LaTeX}$  sceglierà poi l'impaginazione più adeguata.

### 1.2.3 Vantaggi e svantaggi

Quando le persone del mondo WYSIWYG incontrano persone che usano  $\text{\LaTeX}$ , spesso discutono dei “vantaggi di  $\text{\LaTeX}$  rispetto a un normale elaboratore di testi” o viceversa. La cosa migliore da fare quando inizia una simile discussione è tenere un profilo basso poiché tali discussioni sfuggono spesso di mano. Ma a volte non c'è via d'uscita...

Ecco dunque alcuni argomenti. I principali vantaggi di  $\text{\LaTeX}$  rispetto a un normale elaboratore di testi sono i seguenti:

- Sono disponibili delle impaginazioni professionali che veramente rendono un documento come se fosse “stampato”.
- La composizione delle formule matematiche è supportata in modo completamente corrispondente alle necessità.
- Gli utenti devono solo imparare alcuni comandi facili da capire che specificano la struttura logica di un lavoro. Non devono quasi mai arrembiare con la reale impaginazione del documento.
- Anche le strutture complesse come note al piede, riferimenti, indici e bibliografie possono essere generati facilmente.

- Esistono pacchetti aggiuntivi aperti<sup>3</sup> e gratuiti per molte operazioni tipografiche non direttamente supportate dal  $\text{\LaTeX}$  di base. Per esempio, sono disponibili pacchetti per inserire figure in diversi formati, o per comporre bibliografie conformi a un preciso standard. Molti di questi pacchetti sono descritti in *The  $\text{\LaTeX}$  Companion* [3].
- $\text{\LaTeX}$  incoraggia gli autori a scrivere testi ben strutturati perché è così che  $\text{\LaTeX}$  lavora: specificando la struttura.
- $\text{\TeX}$ , il motore di composizione di  $\text{\LaTeX}$  2 $\epsilon$ , è altamente portabile e gratuito. Quindi il sistema gira su quasi ogni piattaforma hardware disponibile.

$\text{\LaTeX}$  ha anche degli svantaggi; ma qui si preferisce riportare le parole stesse di Leslie Lamport alla domanda di un intervistatore:<sup>4</sup>

**Intervistatore:** Three  $\text{\LaTeX}$  mistakes that people should stop making?

**Lamport:** Working too much about format and not enough on content.

– Working too much about format and not enough on content.

– Working too much about format and not enough on content.

Anche se, a rigore, gli errori da evitare non sono svantaggi, la risposta di Lamport spiega quali sono le cose che gli utenti, sbagliando, preferirebbero fare, visto che con i word processor quelle cose si fanno facilmente.

### 1.3 I file di input di $\text{\LaTeX}$

L'input per  $\text{\LaTeX}$  è un file di testo. Su Unix/Linux i file di testo sono piuttosto comuni. Questo file deve contenere sia il testo del documento sia i comandi che dicono a  $\text{\LaTeX}$  come comporlo. Un editor adatto a  $\text{\LaTeX}$  conterrà un programma per creare e modificare file di *input* (“input” d’ora in poi) per  $\text{\LaTeX}$  in formato testuale.

#### 1.3.1 Spazi

I caratteri di spaziatura come lo spazio e la tabulazione sono trattati uniformemente da  $\text{\LaTeX}$  come “spazio”. Più caratteri di spaziatura *consecutivi* sono trattati come *un solo* “spazio”. Lo spazio bianco all’inizio di una riga viene generalmente ignorato e una singola interruzione di riga viene trattata come “spazio”.

<sup>3</sup>“Aperto” riferito a un codice di programmazione si riferisce al diritto dell’utente di esaminarlo e anche di modificarlo, ovviamente sotto certe condizioni esplicitate dall’autore mediante una licenza.

<sup>4</sup>«Quali sono i tre errori  $\text{\LaTeX}$  che gli utenti dovrebbero smettere di fare?» «Dedicarsi troppo alla forma e poco al contenuto.» Da *How (L<sup>A</sup>) $\text{\TeX}$  changed the face of mathematics: An E-interview with Leslie Lamport, the author of  $\text{\LaTeX}$* . TUGboat, 2001, volume 22, pag. 20–22.

Una riga vuota tra due righe di testo definisce la fine di un capoverso. Più righe vuote sono trattate allo stesso modo di *una* riga vuota. Il testo sottostante è un esempio. Sul lato sinistro c'è il testo dell'input e sul lato destro c'è l'output formattato.<sup>5</sup>

```
Non importa se inserite
uno o più spazi
dopo una parola.
```

```
Una riga vuota inizia un nuovo
capoverso.
```

```
Non importa se inserite uno o più spazi
dopo una parola.
```

```
Una riga vuota inizia un nuovo
capoverso.
```

### 1.3.2 Caratteri speciali

I seguenti simboli sono caratteri riservati che in L<sup>A</sup>T<sub>E</sub>X hanno un significato speciale; potrebbero non essere tutti disponibili in ogni font. Digitandoli direttamente nel testo, normalmente non saranno stampati ma piuttosto costringeranno L<sup>A</sup>T<sub>E</sub>X a fare cose non desiderate.

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

Però questi caratteri possono essere usati lo stesso nei documenti prefissandoli con una barra inversa:

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

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

Molti altri simboli possono essere stampati con comandi speciali nelle formule matematiche o come accenti. Il carattere barra inversa `\` *non* può essere inserito preponendogli un'altra barra inversa (`\\`); questa sequenza è usata come interruzione di riga. Si usi piuttosto il comando `\textbackslash`.

### 1.3.3 I comandi di L<sup>A</sup>T<sub>E</sub>X

I comandi di L<sup>A</sup>T<sub>E</sub>X distinguono tra il maiuscolo e il minuscolo e assumono uno dei seguenti formati.

1. Iniziano con una barra inversa `\` seguita da un nome formato di sole lettere. Il nome di un comando è terminato da uno spazio, un numero o qualunque altro carattere diverso da una lettera;
2. Consistono in una barra inversa ed esattamente un solo carattere diverso da una lettera;

<sup>5</sup>Per creare gli esempi come questo, si è creato un apposito ambiente; creare nuovi comandi e nuovi ambienti è una delle particolarità del sistema T<sub>E</sub>X; lo si descriverà meglio nel seguito.

L<sup>A</sup>T<sub>E</sub>X ignora gli spazi dopo i comandi letterali. Se si vuole ottenere uno spazio bianco dopo i comandi bisogna mettere un gruppo vuoto seguito da uno spazio `{ }_L`, o un comando speciale di spaziatura, per esempio `\_L`.

```
I nuovi utenti di \TeX possono
dimenticare gli spazi
dopo un comando. %compone male
```

```
Gli utenti \TeX\ avanzati sono
\TeX nici e sanno come usare
gli spazi. %compone bene
```

I nuovi utenti di T<sub>E</sub>X possono dimenticare gli spazi dopo un comando.

Gli utenti T<sub>E</sub>X avanzati sono T<sub>E</sub>Xnici e sanno come usare gli spazi.

Alcuni comandi richiedono un argomento che gli dev'essere 'passato' tra parentesi graffe `{ }` dopo il proprio nome. Alcuni comandi prendono degli argomenti facoltativi che vanno inseriti dopo il nome del comando tra parentesi quadre `[ ]`.

`\langle comando \rangle [ \langle argomento opzionale \rangle ] { \langle argomento \rangle }`

I prossimi esempi usano alcuni comandi L<sup>A</sup>T<sub>E</sub>X. Non vi si badi; verranno spiegati più avanti.

```
Potete \textsl{appoggiarvi} a me!
```

Potete *appoggiarvi* a me!

```
Per piacere, inizia un nuova riga
esattamente qui!\newline Grazie!
```

Per piacere, inizia un nuova riga esattamente qui!  
Grazie!

### 1.3.4 Commenti

Quando L<sup>A</sup>T<sub>E</sub>X trova un carattere `%` mentre elabora un file di input, ignora il resto della riga corrente, l'interruzione di riga e tutti gli spazi bianchi all'inizio della riga successiva.

Ciò può essere usato per scrivere delle note nell'input che non saranno mostrate nella versione stampata.

```
Questo è uno %stupido
%Miglior: istruttivo <----
esempio: Superca%
         lifragili%
         stichespiralidoso
```

Questo è uno esempio: Supercalifragilistichespiralidoso

Il carattere `%` può anche essere usato per dividere le righe di input lunghe in cui non sono ammessi spazi bianchi o interruzioni di riga.

Per i commenti più lunghi, si può usare l'ambiente `comment` fornito dal pacchetto `verbatim`. Si aggiunga la riga `\usepackage{verbatim}` al preambolo del documento come spiegato più avanti.

```
Questo è un altro
\begin{comment}
abbastanza stupido
ma utile
\end{comment}
esempio per incorporare
commenti nel vostro documento.
```

Questo è un altro esempio per incorporare commenti nel vostro documento.

Attenzione: questo non funzionerà in ambienti complessi come, per esempio, quelli matematici. I commenti con `%`, invece, funzionano sempre.

## 1.4 Struttura degli input

Quando  $\text{\LaTeX} 2_{\epsilon}$  elabora un input, si aspetta che questo abbia una determinata struttura. Ogni input dovrà iniziare col comando

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

Questo specifica il tipo di documento che si intende scrivere. Dopo di ciò, si aggiungono i comandi per modificare lo stile dell'intero documento, o si caricano i pacchetti che aggiungono nuove funzionalità al nucleo di  $\text{\LaTeX}$ . Per caricare uno di questi pacchetti si usa il comando

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

Quando tutto il lavoro di impostazione è fatto,<sup>6</sup> si può incominciare il corpo del testo subito dopo comando

```
\begin{document}
```

Ora si può inserire il testo comprensivo di alcuni comandi  $\text{\LaTeX}$  utili. Alla fine del documento si aggiunge il comando

```
\end{document}
```

che indica a  $\text{\LaTeX}$  la fine delle operazioni. Qualunque cosa segua tale comando sarà ignorata dal programma.

La figura 1.1 mostra il contenuto di un file  $\text{\LaTeX}$  minimo. Un file di input leggermente più complicato è mostrato nella figura 1.2.

---

<sup>6</sup>Tutto quanto compreso tra `\documentclass` e `\begin{document}` è chiamato *preambolo*.

---

```
\documentclass{article}

\begin{document}

Piccolo è bello.

\end{document}
```

---

Figura 1.1: Un file  $\LaTeX$  minimo

---

```
\documentclass[a4paper,11pt]{article}
% definisce il titolo
\author{M.~Rossi}
\title{Minimalismo}

\begin{document}

% genera il titolo
\maketitle

% inserisce l'indice (generale)
\tableofcontents

\section{Alcune parole interessanti}

Bene, e qui comincia il mio simpatico articolo.

\section{Arrivederci mondo}

\ldots\ e qui termina.

\end{document}
```

---

Figura 1.2: Esempio di un articolo di rivista realistico. Si noti che tutti i comandi mostrati in quest'esempio verranno spiegati più avanti nell'introduzione.

## 1.5 Una tipica sessione della riga di comando

Scommetto che si resta sorpresi nel vedere il risultato del conciso file di input di  $\text{\LaTeX}$  mostrato alla pagina 8. Ecco un po' d'aiuto:  $\text{\LaTeX}$  non ha una interfaccia grafica con bottoni carini da cliccare. È solo un programma che macina un input. Alcune installazioni  $\text{\LaTeX}$  hanno un front-end grafico dove c'è un bottone per inviare a  $\text{\LaTeX}$  la compilazione dell'input. Su altri sistemi può essere necessaria un po' di digitazione; ecco dunque come costringere  $\text{\LaTeX}$  a compilare l'input su un sistema senza interfaccia grafica. Si noti che la seguente descrizione suppone che sia già presente nel computer col quale si sta lavorando un'installazione del sistema  $\text{\TeX}$  funzionante.

1. Si crei il file di input per  $\text{\LaTeX}$ . Questo file oggi risulta scritto in codice UTF-8 (Unicode), ma per farlo si usi un editor di puro testo, per esempio su Windows si può usare *Notepad++* (da installare); su Mac si può usare *TextEdit.app*; sulle piattaforme Linux si può usare *Vim*. Quando si salva il file, ci si assicuri che il nome sia completo con l'estensione `.tex`.
2. Si apra un terminale o una finestra-comandi, si esegua `cd` fino a portarsi nella cartella in cui si trova l'input e si esegua il comando `pdflatex <nome del file>` senza specificare l'estensione. Se la compilazione andrà a buon fine, si otterrà un file `.pdf`. Può essere necessario eseguire `pdf\LaTeX` diverse volte per avere il sommario e tutti i riferimenti interni corretti. Quando l'input contiene un errore, `pdf\LaTeX` ne informerà l'utente e interromperà la compilazione dell'input. Si digiti `H + Invio` per avere una descrizione dell'errore, poi si digiti `X + Invio` per terminare il processo e tornare alla riga di comando.

## 1.6 La struttura del documento

### 1.6.1 Classi di documento

La prima informazione che  $\text{\LaTeX}$  ha bisogno di conoscere quando elabora un file sorgente è il tipo di documento che l'autore vuole creare. Ciò è specificato col comando `\documentclass`.

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

Qui `<classe>` specifica il tipo di documento da creare. La tabella 1.1 elenca le classi di documento spiegate in questa introduzione. La distribuzione  $\text{\LaTeX} 2_{\epsilon}$  fornisce classi aggiuntive per altri tipi di documenti, incluse le lettere e le presentazioni. L'argomento `<opzioni>` personalizza il comportamento della classe. Le opzioni devono essere separate da virgole. Le opzioni più comuni per le classi di documento standard sono elencate nella tabella 1.2.

Esempio: Un input per L<sup>A</sup>T<sub>E</sub>X potrebbe iniziare con la riga

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

che istruisce L<sup>A</sup>T<sub>E</sub>X a comporre il documento come un articolo con una dimensione di base del font di *undici punti* per produrre un impaginato adatto alla stampa *fronte-retro* (*doppia faccia*) su *fogli A4*.

### 1.6.2 Pacchetti

Mentre si scrive il documento, probabilmente si troveranno delle aree in cui il nucleo di L<sup>A</sup>T<sub>E</sub>X non può risolvere un dato problema. Se si vuole includere della grafica, del testo a colori o del codice sorgente da un file nel documento, è necessario estendere le capacità del programma. Tali estensioni sono chiamate *pacchetti*. Ogni pacchetto è invocato col comando

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

dove *pacchetto* è il nome del pacchetto e *opzioni* è una lista di parole chiave che attivano caratteristiche speciali nel pacchetto. Il comando `\usepackage` va nel preambolo del documento. Ci si riferisca alla sezione 1.4 per i dettagli.

Alcuni pacchetti vengono forniti con la distribuzione di base di L<sup>A</sup>T<sub>E</sub>X (vedi la tabella 1.3). Altri sono forniti separatamente. Si possono trovare ulteriori informazioni sui pacchetti installati nella *Local Guide* [5].

Tabella 1.1: Classi di documento

---

<b>article</b>	per articoli di riviste, presentazioni, brevi relazioni, documentazione di programmi, inviti, ...
<b>book</b>	per libri veri e propri.
<b>report</b>	per relazioni lunghe contenenti diversi capitoli, brevi libri, tesi, ...
<b>minimal</b>	è veramente minimale. Imposta solo una dimensione di pagina e un font di base. È usata principalmente per scopi di collaudo.
<b>proc</b>	classe per gli atti di un congresso; è basata sulla classe article.
<b>slides</b>	per documenti da proiettare su schermo. La classe usa grandi caratteri lineari molto adatti a questo tipo di documenti. È meglio prendere in considerazione l'uso alternativo della classe Beamer.

---

Tabella 1.2: Opzioni per la classe di documento

---

<code>10pt</code> , <code>11pt</code> , <code>12pt</code>	Imposta la dimensione del font principale nel documento. Se non è specificata alcuna opzione, viene assunta <code>10pt</code> .
<code>a4paper</code> , <code>letterpaper</code> , ...	Definisce la dimensione del foglio. La dimensione preimpostata può essere <code>a4paper</code> o <code>letterpaper</code> a seconda di come si è configurata l'installazione del sistema <code>T<sub>E</sub>X</code> . Oltre a queste, possono essere specificate <code>a5paper</code> , <code>b5paper</code> , <code>executivepaper</code> e <code>legalpaper</code> .
<code>fleqn</code>	Compone le formule fuori testo allineate a sinistra anziché centrate.
<code>leqno</code>	Pone la numerazione delle formule sul lato sinistro anziché su quello destro.
<code>titlepage</code> , <code>notitlepage</code>	Specifica se dopo il titolo del documento deve o non deve essere iniziata una nuova pagina. La classe <code>article</code> è preimpostata per non iniziarla, mentre <code>report</code> e <code>book</code> sì.
<code>onecolumn</code> , <code>twocolumn</code>	Istruiscono <code>L<sup>A</sup>T<sub>E</sub>X</code> a comporre il documento su una colonna ( <code>onecolumn</code> ) o due colonne ( <code>twocolumn</code> ).
<code>twoside</code> , <code>oneside</code>	Specificano se il documento dev'essere stampato fronte-retro o solo fronte. Le classi <code>article</code> e <code>report</code> sono preimpostate per la stampa solo fronte ( <i>one sided</i> ) mentre la classe <code>book</code> è predisposta per la stampa fronte-retro. ( <i>two sided</i> ). Si noti che questa opzione concerne solamente lo stile del documento. L'opzione <code>twoside non</code> dice alla stampante di stampare veramente un documento fronte-retro.
<code>landscape</code>	Cambia l'orientamento del documento per stamparlo in modalità panoramica (orizzontale).
<code>openright</code> , <code>openany</code>	Fanno in modo che i capitoli inizino o solo sulle pagine di destra o alla prima pagina disponibile rispettivamente. Questo non funziona con la classe <code>article</code> poiché questa non ha i capitoli. La classe <code>report</code> è preimpostata per iniziare i capitoli alla prima pagina disponibile e la classe <code>book</code> li inizia sulle pagine di destra.

---

La sorgente primaria di informazioni su pacchetti di  $\text{\LaTeX}$  è *The  $\text{\LaTeX}$  Companion* [3]. Questo testo contiene le descrizioni di centinaia di pacchetti insieme alle informazioni su come creare le estensioni personali per  $\text{\LaTeX} 2_{\epsilon}$ .

Le moderne distribuzioni di  $\text{\TeX}$  hanno un enorme numero di pacchetti preinstallati. Si usi il comando `texdoc` per accedere alla documentazione di un pacchetto che sia in formato PDF o in formato testo. Qualche pacchetto contiene solo il file sorgente della sua documentazione; esso ha estensione `.dtx` e lo si può leggere con qualunque editor di testo.

### 1.6.3 Stili della pagina

$\text{\LaTeX}$  fornisce tre combinazioni predefinite di testatina/piedino, conosciute anche come stili di pagina. Il parametro  $\langle stile \rangle$  del comando `\pagestyle` indica quale stile di pagina usare: `empty` (senza testatina né piedino), `plain` (col solo piedino), `headings` (con testatina e piedino). La tabella 1.4 elenca gli stili di pagina predefiniti.

```
\pagestyle{\langle stile \rangle}
```

definisce quale usare. In realtà esisterebbe anche lo stile `myheadings`; funziona come `headings`, ma senza l'automatismo di quest'ultimo; che cosa scrivere nelle testatine va specificato dall'utente.

È possibile cambiare stile della pagina corrente col comando

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

Si può trovare una descrizione di come personalizzare testatine e piedini nella sezione 4.3 nella pagina 82.

## 1.7 Altri file

Quando si lavora con  $\text{\LaTeX}$ , ci si troverà presto in un labirinto di file con varie estensioni e probabilmente senza nessun indizio.<sup>7</sup> La lista che segue spiega i vari tipi di file che si possono incontrare lavorando con  $\text{\LaTeX}$ . Si prega di notare che questo elenco non pretende di essere completo, ma se si trova un'estensione mancante, ritenuta però importante, per piacere se ne informino i curatori di questo testo tramite il forum del  $\text{\CTAN}$ .

<sup>7</sup>Alcuni sistemi operativi si affidano completamente alle icone che sono associate a ciascuna estensione; l'estensione viene mostrata solo per i file cui non è associato nessun programma per elaborarlo. Si raccomanda vivamente di configurare il programma che elenca i file in ogni cartella (`Explorer`, `Navigator`, `Finder`, eccetera) in modo che ne mostri sempre l'estensione.

Tabella 1.3: Alcuni dei pacchetti distribuiti con L<sup>A</sup>T<sub>E</sub>X

---

<code>doc</code>	Permette di documentare i programmi di L <sup>A</sup> T <sub>E</sub> X. Descritto in <code>doc.pdf</code> e in <i>The L<sup>A</sup>T<sub>E</sub>X Companion</i> [3].
<code>exscale</code>	Fornisce le versioni scalate del font matematici estensibili. Descritto in <code>ltxscale.pdf</code> .
<code>fontenc</code>	Specifica quale codifica dei font L <sup>A</sup> T <sub>E</sub> X dovrà usare. Descritto in <code>fntguide.pdf</code> .
<code>ifthen</code>	Fornisce i comandi della forma “se... allora fa’... altrimenti fa’...” Descritto in <code>ifthen.pdf</code> e <i>The L<sup>A</sup>T<sub>E</sub>X Companion</i> [3].
<code>latexsym</code>	Per accedere al font di alcuni simboli di L <sup>A</sup> T <sub>E</sub> X (obsoleti) si potrebbe usare il pacchetto <code>latexsym</code> . Descritto in <code>latexsym.pdf</code> e in <i>The L<sup>A</sup>T<sub>E</sub>X Companion</i> [3].
<code>makeidx</code>	Fornisce i comandi per produrre gli indici analitici. Sarebbe meglio usare il pacchetto <code>imakeidx</code> come descritto nella nota 2 della sezione 4.2. Descritto in <code>imakeidx.pdf</code> ; <i>The L<sup>A</sup>T<sub>E</sub>X Companion</i> [3] si riferisce ancora solo a <code>makeidx</code> .
<code>syntonly</code>	Elabora un documento senza comporlo. Serve per controllare la sintassi del mark up e per trovarne gli eventuali errori.
<code>inputenc</code>	Permette di specificare una codifica dell’input fra UTF-8, ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows o una codifica definita dall’utente. Oggi si dovrebbe usare solo la codifica UTF-8. Descritto in <code>inputenc.pdf</code> .

---

Tabella 1.4: Gli stili di pagina predefiniti di  $\text{\LaTeX}$ .

---

**plain** stampa i numeri di pagina in fondo alla pagina, al centro del piedino. Questo è lo stile di pagina usato in mancanza di indicazioni diverse.

**headings** scrive nella testatina di ogni pagina il titolo del capitolo corrente o quello della sezione corrente, affiancati dal numero di pagina, mentre il piedino rimane vuoto. (Questo è lo stile usato in questo documento.)

**empty** lascia vuote sia le testatine sia i piedini.

---

**.tex** file di input di  $\text{\LaTeX}$ . Può essere compilato con uno dei diversi programmi di composizione del sistema  $\text{\TeX}$  basati sul linguaggio  $\text{\LaTeX}$ .

**.sty** Pacchetto di estensione per  $\text{\LaTeX}$ . Lo si invoca mediante il comando `\usepackage`.

**.dtx** *Documented*  $\text{\TeX}$ . Questo è il principale formato di distribuzione per i file di estensione di  $\text{\LaTeX}$ . Se si elabora un file `.dtx` si otterrà il codice documentato delle macro del pacchetto  $\text{\LaTeX}$  contenuto nel file `.dtx`.

**.ins** L'installatore per i file contenuti nel corrispondente file `.dtx`. Se si scarica un pacchetto  $\text{\LaTeX}$  dalla rete, si disporrà normalmente un file `.dtx` e un file `.ins`. Si esegua `pdf\LaTeX` sul file `.ins` per estrarre quei file che il file `.dtx` contiene e documenta.

**.cls** I file di classe definiscono quale aspetto generale avrà il documento. Sono selezionati col comando `\documentclass`.

**.fd** File di descrizione di font che informa  $\text{\LaTeX}$  delle caratteristiche dei font da usare per la composizione.

I file seguenti sono generati quando si esegue  $\text{\LaTeX}$  sul file di input:

**.dvi** Device Independent File (file indipendente dal dispositivo). Questo è il risultato principale di una classica compilazione con  $\text{\TeX}$  o  $\text{\LaTeX}$ . Si visualizza questo contenuto con un programma di visualizzazione di file in formato DVI che di solito è anche in grado di inviarlo alla stampante. Oggi è raro incontrare questo tipo di file perché, se si sta usando `pdf\LaTeX` o uno degli altri programmi di composizione moderni, si ottengono file con l'estensione `.pdf`.

**.log** Fornisce un rapporto dettagliato di cos'è successo durante l'ultima compilazione.

- .toc** Immagazzina tutti i titoli creati dai comandi di sezionamento. Viene letto nella successiva compilazione e usato per produrre il sommario o indice generale.
- .lof** È come il `.toc`, ma per l'elenco delle figure.
- .lot** È ancora simile, ma per l'elenco delle tabelle.
- .aux** Un altro file che veicola informazioni da una compilazione all'altra. In aggiunta ad altre cose, il file `.aux` serve per mantenere le informazioni associate ai riferimenti incrociati e alle citazioni bibliografiche.
- .idx** Se il documento contiene un indice analitico,  $\text{\LaTeX}$  stiva tutte le parole che lo compongono in questo file. Si elabora questo file con `makeindex`. Ci si riferisca alla sezione 4.2 alla pagina 80 per maggiori informazioni sull'indicizzazione.
- .ind** Il file `.idx` elaborato, pronto per l'inclusione nel documento al successivo ciclo di compilazione.
- .ilg** File di log contenente quanto fatto da `makeindex`.

## 1.8 Grandi progetti

Lavorando su grandi documenti, potrebbe essere utile separare il file di input in più parti.  $\text{\LaTeX}$  ha due metodi che aiutano a farlo.

Il primo ricorre a due comandi: `\include` e `\includeonly`. Il primo comando ha la sintassi seguente:

`\include{<nome del file>}`

Lo si usi nel corpo del documento per inserirvi il contenuto di un altro file nominato `<nome del file>.tex`. Si noti che  $\text{\LaTeX}$  inizierà una nuova pagina prima di elaborare il materiale contenuto in `<nome del file>.tex`.

Il secondo comando può essere usato nel preambolo. Permette di istruire  $\text{\LaTeX}$  affinché includa solo i file i cui nomi sono contenuti nella lista specificata come argomento di `\includeonly`:

`\includeonly{<nome file_1>,<nome file_2>,...}`

Dopo l'esecuzione di questo comando nel preambolo del documento, saranno eseguiti solo gli `\include` relativi ai nomi dei file elencati nell'argomento di `\includeonly`.

Siccome il comando `\include` inizia a comporre il testo incluso su una nuova pagina, questo procedimento è utile quando si usa `\includeonly` per includere interi capitoli, perché le interruzioni di pagina non varieranno anche

qualora alcuni file da includere siano omessi. A volte ciò potrebbe non essere desiderabile. In questo caso si usi il secondo metodo che prevede il comando

```
\input{<nome del file>}
```

Questo include semplicemente il file specificato. Niente effetti speciali, nessuna conseguenza.

Per far sì che  $\text{\LaTeX}$  controlli rapidamente il documento, si usi il pacchetto `syntonly`. Questo fa fare a  $\text{\LaTeX}$  un semplice controllo sulla correttezza della sintassi e dell'uso dei comandi ma non produce alcun file di uscita. Poiché  $\text{\LaTeX}$  gira più velocemente in questa modalità, si può risparmiare del tempo prezioso. L'uso è molto semplice:

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

Quando si vuole produrre davvero il documento, si commenti semplicemente la seconda riga (aggiungendo all'inizio un segno `%`).

## Capitolo 2

# Comporre il testo

Dopo aver letto il capitolo precedente, si dovrebbero conoscere gli elementi fondamentali di cui è composto un documento  $\text{\LaTeX}2_{\epsilon}$ . In questo capitolo completeremo la rimanente struttura che sarà necessario conoscere per produrre davvero documenti stampabili. Con vari esempi si mostreranno i dettagli più importanti.

### 2.1 La struttura del testo e della lingua<sup>1</sup>

Lo scopo principale nella scrittura di un testo è quello di trasmettere idee, informazioni o conoscenze al lettore. Il lettore lo capirà meglio se queste idee sono ben strutturate, e vedrà e percepirà questa struttura molto meglio, se la forma tipografica riflette la struttura logica e semantica del contenuto.

$\text{\LaTeX}$  è diverso da altri sistemi di composizione, in quanto di un testo bisogna “dirgli” solo la struttura logica e semantica. Il programma poi deriva la forma tipografica in base alle “regole” contenute nel file di classe del documento e in vari file di estensione.

La più importante unità testuale in  $\text{\LaTeX}$  (e in tipografia) è il capoverso. Lo chiamiamo “unità testuale” perché un capoverso è la forma tipografica che dovrebbe riflettere un pensiero coerente o un’idea. Nelle prossime sezioni si imparerà come forzare un’interruzione di riga con `\`, per esempio, e come realizzare un’interruzione di capoverso lasciando una riga vuota nel codice sorgente. Pertanto, se si incomincia un nuovo pensiero, dovrebbe iniziare un nuovo capoverso e, in caso contrario, dovrebbero essere utilizzate solo interruzioni di riga. In caso di dubbio sulle interruzioni di capoverso, si pensi al testo come a un veicolo di idee e pensieri. Se si ha un’interruzione di capoverso, ma il vecchio pensiero continua, dovrebbe essere rimossa. Se nello stesso capoverso si avesse qualche riga di un pensiero totalmente nuovo, allora dovrebbe essere interrotto.

---

<sup>1</sup>Questa sezione originariamente è stata composta da Hanspeter Schmid.

La maggior parte delle persone sottovaluta completamente l'importanza di interruzioni di capoverso ben posizionate. Molte di esse non sanno nemmeno quale sia il loro significato o, specialmente in  $\text{\LaTeX}$ , ne introducono senza saperlo. Quest'ultimo errore è particolarmente facile da commettere se nel testo vengono adoperate delle equazioni. Si guardino i seguenti esempi e si scopra perché a volte vengono lasciate righe vuote (interruzioni di capoverso) prima e dopo l'equazione, e talvolta no. (Se ancora non si capiscono tutti i comandi abbastanza bene da comprendere questi esempi, si legga questo capitolo e quello seguente, quindi si torni a leggere di nuovo questa sezione.)

```
% Esempio 1
\ldots Einstein introdusse la sua
formula
\begin{equation}
E = m \cdot c^2
\end{equation}
che è allo stesso tempo la formula
fisica più conosciuta e la
meno compresa.
```

... Einstein introdusse la sua formula

$$E = m \cdot c^2 \quad (2.1)$$

che è allo stesso tempo la formula fisica più conosciuta e la meno compresa.

```
% Esempio 2
\ldots da cui deriva la legge delle
correnti di Kirchhoff:
\begin{equation}
\sum_{k=1}^n I_k = 0
\end{equation}
Si può ricavare la legge delle
tensioni di Kirchhoff\ldots
```

... da cui deriva la legge delle correnti di Kirchhoff:

$$\sum_{k=1}^n I_k = 0 \quad (2.2)$$

Si può ricavare la legge delle tensioni di Kirchhoff...

```
% Esempio 3
\ldots che ha molti vantaggi.
\begin{equation}
I_D = I_F - I_R
\end{equation}
è la base di un modello di transistoro
molto diverso.
```

... che ha molti vantaggi.

$$I_D = I_F - I_R \quad (2.3)$$

è la base di un modello di transistoro molto diverso.

L'unità di testo successiva più piccola è l'enunciato. Nei testi in inglese, dopo il punto fermo che termina un enunciato c'è uno spazio più grande di quello presente dopo un'abbreviazione.  $\text{\LaTeX}$  tenta di capire da solo quale si desidera avere. Se  $\text{\LaTeX}$  si sbaglia, bisogna dirgli quale dei due si vuole, come spiegato più avanti in questo stesso capitolo.

La strutturazione del testo si estende anche a parti di enunciati. La maggior parte delle lingue ha regole di punteggiatura molto complicate, ma in molte di esse (tra cui tedesco e inglese), si avranno quasi tutte le virgole

al posto giusto, basate sulla breve interruzione che si fa ad ogni virgola quando si legge ad alta voce. In italiano le virgole vanno inserite in modo da sezionare ogni enunciato secondo una logica, non secondo tratti prosodici come per l'inglese o il tedesco; servono per certi incisi, per certe elencazioni, per separare alcune frasi subordinate, ma non servono per respirare, anche se leggendo ad alta voce talvolta lo si fa.

Infine, i capoversi di un testo dovrebbero essere strutturati logicamente anche a un livello superiore, inserendoli in capitoli, paragrafi, sottoparagrafi e così via. Tuttavia, l'effetto tipografico di scrivere `\section{La struttura del testo e della lingua}`, per esempio, è così ovvio, che il modo in cui queste strutture di alto livello dovrebbero essere adoperate si spiega quasi da sé.

Queste strutture gerarchiche, capitolo, paragrafo, eccetera, si dovrebbero usare solo se dentro un capitolo ci sono almeno due paragrafi; se dentro un paragrafo ci sono almeno due sottoparagrafi; eccetera.

## 2.2 Interruzione di riga e di pagina

### 2.2.1 Capoversi giustificati

Spesso i libri sono composti con righe tutte della medesima lunghezza.  $\text{\LaTeX}$  inserisce le necessarie interruzioni di riga e gli spazi tra le parole ottimizzando il contenuto di un intero capoverso. Se necessario, inoltre, esegue la “cesura”<sup>2</sup> delle parole che non starebbero comodamente su una riga. Il modo in cui i capoversi sono composti dipende dalla classe di documento. Di solito, la prima riga di un capoverso è rientrata, e tra due capoversi non viene inserito ulteriore spazio. Per maggiori informazioni, si veda il paragrafo 6.3.2.

In situazioni molto particolari e rare, potrebbe essere necessario ordinare a  $\text{\LaTeX}$  di interrompere una riga:

```
\ \ oppure \newline
```

incomincia una nuova riga senza incominciare un nuovo capoverso.

Il comando

```
\ \*
```

inoltre, proibisce un'interruzione di pagina dopo l'interruzione di riga forzata.

<sup>2</sup>In seno al gruppo di lavoro del TUG sulla sillabazione si distingue fra *syllabification* e *hyphenation*, cioè fra sillabazione e cesura; il primo termine si riferisce alla grammatica e/o alla fonetica; il secondo alla tipografia; le regole tipografiche non violano le regole grammaticali, mentre non vale il viceversa: grammatica: i-de-a; tipografia: idea.

Il comando

```
\newpage
```

incomincia una nuova pagina. I comandi

```
\linebreak[⟨n⟩]
\nolinebreak[⟨n⟩]
\pagebreak[⟨n⟩]
\nopagebreak[⟨n⟩]
```

suggeriscono punti del documento in cui un'interruzione può o non può essere messa. Essi permettono all'utente di influenzare le loro azioni mediante l'argomento opzionale  $\langle n \rangle$ , che può essere impostato su un numero compreso tra zero e quattro. Impostando  $\langle n \rangle$  a un valore inferiore a 4, si lascia a  $\text{\LaTeX}$  la possibilità di ignorare il comando se il risultato apparisse molto brutto.<sup>3</sup>

Non si confondano questi comandi “break” con i comandi “new”. Anche quando si dà un comando “break”,  $\text{\LaTeX}$  prova ancora a pareggiare il bordo destro della riga con la giustezza totale del blocco di testo, come descritto nella prossima sezione; questo può portare a sgradevoli lacune nel testo. Se si vuole veramente incominciare una “nuova riga” o una “nuova pagina”, si adoperi il comando corrispondente: quale sarà il suo nome?

$\text{\LaTeX}$  cerca sempre di produrre le migliori interruzioni di riga possibili. Se non riesce a trovare un modo per spezzare le righe in un modo che soddisfi i suoi elevati standard, può lasciare un segno sulla parte destra del capoverso.  $\text{\LaTeX}$  si lamenta (“overfull hbox”) durante l'elaborazione del file di input. Questo accade molto spesso quando non riesce a trovare un luogo adatto per eseguire la cesura di una parola.<sup>4</sup>

Si può dire a  $\text{\LaTeX}$  di abbassare leggermente i propri standard dando il comando `\sloppy`. Esso previene le righe troppo lunghe aumentando la spaziatura tra le parole—anche se l'output finale non è ottimale. In questo caso viene dato all'utente un avvertimento (“underfull hbox”). Nella maggior parte dei casi, il risultato non sembra molto buono. Il comando `\fussy` riporta  $\text{\LaTeX}$  al proprio comportamento predefinito.

Per applicare l'impostazione `\sloppy` a un solo capoverso, lo si può richiudere nell'ambiente `sloppypar`; dopo questo ambiente non è necessario specificare `\fussy`.

<sup>3</sup>La bruttezza, secondo  $\text{\LaTeX}$ , è il risultato di un calcolo matematico piuttosto complesso; quanto più alto è il valore della bruttezza, tanto più opportuna diventa un'interruzione di riga o di pagina. Il lettore non si spaventi per quei calcoli complicati; non li deve mai fare, perché ci pensa  $\text{\LaTeX}$  per conto proprio.

<sup>4</sup>Anche se  $\text{\LaTeX}$  dà un avvertimento quando ciò accade (`Overfull hbox`) e visualizza il numero della riga incriminata, tali righe non sono sempre facili da trovare. Se si imposta l'opzione `draft` nel comando `\documentclass`, queste righe saranno contrassegnate con una spessa linea nera sul margine destro.

### 2.2.2 Sillabazione e cesura

L<sup>A</sup>T<sub>E</sub>X divide in sillabe le parole in fine di riga con regole più severe di quelle grammaticali; sarebbe bene riservare la parola ‘sillabazione’ per le regole grammaticali, e ‘cesura’ per le divisioni tipografiche.

L<sup>A</sup>T<sub>E</sub>X esegue la cesura delle parole ogni volta che serve. Se l’algoritmo non trova i punti di cesura corretti, si provveda a mano usando i seguenti comandi per indicare a T<sub>E</sub>X l’eccezione.

Il comando

```
\hyphenation{lista di parole}
```

fa sì che la cesura delle parole elencate nel proprio argomento avvenga solo nei punti contrassegnati da “-”. L’argomento del comando dovrebbe contenere solo parole composte da lettere normali, o piuttosto segni che vengono considerati lettere normali da L<sup>A</sup>T<sub>E</sub>X; per esempio, per quel che riguarda l’italiano, l’apostrofo non è una lettera, ma ai fini della cesura è trattato come se lo fosse. Le cesure particolari vengono memorizzate per la lingua attiva quando si verifica il comando di sillabazione. Ciò significa che se si inserisce con `\hyphenation` una lista di cesure particolari nel preambolo del documento, questo influenzerà la sillabazione della lingua inglese (la lingua predefinita). Se si posiziona il comando dopo `\begin{document}` e si sta adoperando un pacchetto per il supporto della lingua nazionale come `babel` o `polyglossia`, allora le cesure particolari saranno attive nella lingua selezionata mediante i comandi specifici di quei pacchetti.

L’esempio seguente permetterà la cesura di “transalpino” e “Transalpino”, e la impedirà per “FORTRAN”, “Fortran” e “fortran”. Nell’argomento non sono ammessi caratteri speciali o simboli.

Esempio:

```
\hyphenation{FORTRAN trans-al-pi-no}
```

Il comando `\-` marca un possibile punto di cesura in una parola. Questo diventa anche l’unico punto in cui in questa parola la cesura è consentita. Questo comando è particolarmente utile per le parole che non contengono solo lettere (per esempio caratteri accentati, se non si usa il pacchetto `inputenc`). Oggi la codifica UTF-8 è quella di default e quindi l’uso di `\-` diventa utile solo per parole straniere o con radici straniere rispetto alla lingua principale.

Le zecche trasmettono la malattia  
leish\maniosi.

Penso che questo sia: su\per\ca\-%  
li\fra\gi\li\stich\-e\spi\-%  
ra\li\do\so

Le zecche trasmettono la malattia leish-  
maniosi.

Penso che questo sia: supercalifragili-  
stichesimalidoso

Più parole possono essere tenute insieme su una riga con il comando

```
\mbox{⟨testo⟩}
```

Esso in scatola il  $\langle testo \rangle$  in modo che sia tenuto insieme in ogni circostanza. Nell'esempio che segue la mancanza della cesura, dovuta alla scatola, si riflette sull'aumento dello spazio interparola.

Presto il numero di telefono  
diventerà `\mbox{0116 291 2319}`.

`\noindent` Presto il numero di telefono  
diventerà 0116 291 2319.

```
Presto il numero di telefono diventerà  
0116 291 2319.
```

```
Presto il numero di telefono diventerà 0116  
291 2319.
```

`\fbox` è simile a `\mbox`, ma in più si vedrà una cornice disegnata attorno al contenuto.

## 2.3 Stringhe preconfezionate

In alcuni esempi nelle pagine precedenti si sono visti alcuni comandi  $\text{\LaTeX}$  molto semplici per la composizione di stringhe di testo speciali:

Comando	Esempio	Descrizione
<code>\today</code>	30 novembre 2020	Data corrente
<code>\TeX</code>	$\text{\TeX}$	Il compositore preferito
<code>\LaTeX</code>	$\text{\LaTeX}$	Il nome del gioco
<code>\LaTeXe</code>	$\text{\LaTeX} 2_{\epsilon}$	La versione corrente

## 2.4 Caratteri speciali e simboli

### 2.4.1 Virgolette

*Non* si deve adoperare " per le virgolette come si farebbe con una macchina per scrivere. Nell'editoria esistono speciali virgolette di apertura e di chiusura. In  $\text{\LaTeX}$  si possono adoperare due ` (accento grave) per le virgolette di apertura e due ' (virgoletta verticale) per quelle di chiusura; evidentemente è errato inserire due '' in apertura di un inciso virgolettato. Per le virgolette singole, se ne adoperi solo un segno ‘ in apertura e un solo ’ in chiusura.. Lavorando con la codifica UTF-8 (quella di default con gli strumenti di composizione del sistema  $\text{\TeX}$ ), *tastiera permettendo*, si possono immettere nel file sorgente direttamente i segni “ e ”.

```
“Premi il tasto ‘X’, per favore”
```

```
“Premi il tasto ‘X’, per favore”
```

Si è evidenziato *tastiera permettendo*; le tastiere nazionali generalmente contengono i tasti con le virgolette nazionali, e consentono l'uso di combinazioni di tasti per i segni non riportati sui tasti. La tastiera italiana è forse l'unica tastiera al mondo che permette di inserire segni che non appartengono all'italiano, per esempio 'ç', ma non contiene né l'accento grave, né le lettere maiuscole accentate; generalmente si ricorre alle vecchie abitudini della macchina da scrivere, ma E' è sbagliato, mentre è giusto scrivere È; le macro di T<sub>E</sub>X vengono in aiuto, ma resta sempre il problema che manca l'accento grave. Meno male che quasi tutti gli editor permettono di inserire i segni mancanti ricorrendo alla tabella dei caratteri.

### 2.4.2 Tratti, trattini e lineette

L<sup>A</sup>T<sub>E</sub>X conosce quattro tipi di tratti. Si ottengono tre di essi con un diverso numero di trattini consecutivi. Il quarto segno, per la verità, non è affatto un trattino, ma è il segno matematico meno: -, −, —, −.

```
Aix-en-Provence, tosco-emiliano\
pagine 13--67\
si--o no? \
$0$, $1$ e $-1$
```

```
Aix-en-Provence, tosco-emiliano
pagine 13-67
si-o no?
0, 1 e -1
```

I nomi per questi segni sono: '-' trattino, '−' tratto, '—' lineetta, e '−' meno.

### 2.4.3 Tilde (~)

Un carattere che si vede spesso negli indirizzi Internet è la tilde. Per produrla in L<sup>A</sup>T<sub>E</sub>X, se si adopera `\~{}`, il risultato (~) non è davvero il meglio. Si usi `\textasciitilde`, invece:

```
http://www.puli.edu/\~{r}emo \
http://www.lave.edu/%
\textasciitilde ranno
```

```
http://www.puli.edu/~remo
http://www.lave.edu/~ranno
```

### 2.4.4 Barra obliqua (/)

Per stampare una barra obliqua tra due parole, per esempio, si può scrivere semplicemente `read/write`, ma questo fa sì che L<sup>A</sup>T<sub>E</sub>X consideri le due parole come una sola e non ne esegua la cesura perché contiene un carattere diverso da una lettera; quindi potrebbero generarsi errori 'overfull'. Per risolvere il problema, si adoperi `\slash`. Per esempio, la scrittura `'read\slash write'` permette la cesura. Ma il normale carattere '/' può ancora essere adoperato per rapporti di numeri o di unità di misura.

`$3/5$ \quad 50\,km/h`

3/5 50 km/h

Nell'esempio, il comando `\,` inserisce uno spazio fine fra le due stringhe che unisce, ma impedisce di andare a capo prima della seconda stringa; in questo modo i simboli delle unità di misura non si staccano dal valore numerico; alcuni usano la tilde che, oltre a vietare il fine riga, inserisce un normale spazio interparola, generalmente troppo grande per separare misura e unità di misura che assieme esprimono un unico concetto.

### 2.4.5 Simbolo di grado (°)

Stampare il simbolo di grado con le funzionalità del nucleo di L<sup>A</sup>T<sub>E</sub>X richiede un piccolo trucco:

Sono `$-30\,^{\circ}\mathrm{C}$` e tra poco diventerò un superconduttore.

Sono  $-30^{\circ}\text{C}$  e tra poco diventerò un superconduttore.

Il pacchetto `textcomp` rende il simbolo di grado disponibile anche come `\textdegree` o, in combinazione con la `C`, adoperando `\textcelsius`.

`30\,\textcelsius` sono  
`86\,\textdegree{F}`.  
 Un angolo di `90\textdegree`.  
 Un angolo di `$25^{\circ} 37'12''$`.

$30^{\circ}\text{C}$  sono  $86^{\circ}\text{F}$ .  
 Un angolo di  $90^{\circ}$ .  
 Un angolo di  $25^{\circ}37'12''$ .

Per i gradi sessagesimali non bisogna lasciare lo spazio fra il valore e il simbolo di grado.

### 2.4.6 Il simbolo dell'euro (€)

Oggi, quando si scrive di soldi, bisogna disporre del simbolo dell'euro. Molti font correnti contengono il simbolo dell'euro. Dopo aver caricato il pacchetto `textcomp` nel preambolo del documento

```
\usepackage{textcomp}
```

si usi il comando

```
\texteuro
```

per accedervi.

Se un font non fornisce il suo simbolo dell'euro o se non piace quello contenuto nel font, si hanno ancora due possibilità:

Prima di tutto, il pacchetto `eurosym`. Esso fornisce il simbolo dell'euro ufficiale:

```
\usepackage[official]{eurosym}
```

Se si preferisce un simbolo dell'euro che corrisponda al font prescelto, si usi l'opzione `gen` invece di `official`.

Tabella 2.1: Una borsa piena di simboli dell'euro

LM + <code>textcomp</code>	<code>\texteuro</code>	€	€	€
<code>eurosym</code>	<code>\euro</code>	€	€	€
<code>[gen]eurosym</code>	<code>\euro</code>	€	€	€

### 2.4.7 Ellissi (...)

In una macchina per scrivere, una virgola o un punto occupano lo stesso spazio di ogni altra lettera. Nella stampa di un libro, questi caratteri occupano solo un piccolo spazio e sono messi molto vicino alla lettera precedente. Tuttavia, inserire un 'ellissi' semplicemente digitando tre punti produrrebbe un risultato sbagliato. Invece, c'è un comando speciale per questi punti. È chiamato

```
\ldots (punti bassi)
```

```
Non così... ma così:\ldots\  
New York, Tokyo, Budapest,\ldots
```

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

### 2.4.8 Legature

Alcune combinazioni di lettere vengono composte non semplicemente mettendo le diverse lettere una dopo l'altra, ma adoperando effettivamente simboli speciali.

`ff fi fl ffi ffl ...` invece di `ff fi fl ffi ffl ...`

Queste cosiddette 'legature' possono essere disabilitate inserendo un comando `\/` tra le due lettere in questione; tale comando è da usare quando due lettere, a causa della loro forma o inclinazione potrebbero scontrarsi:

(f) e (f\)\  
 Il simbolo del femtofarad va  
 scritto così: f\F e non così fF.

(f) e (f)  
 Il simbolo del femtofarad va scritto così:  
 fF e non così fF.

Ciò potrebbe essere necessario con parole composte di due parole (l'esempio è in inglese perché non siamo riusciti a trovare una parola composta italiana formata da due parti di cui la prima finisce con 'f' e la seconda cominci di nuovo con 'f').<sup>5</sup>

\Large Not shelfful but shelf\ful

Not shelfful but shelfful

### 2.4.9 Accenti e caratteri speciali

L<sup>A</sup>T<sub>E</sub>X consente l'uso di accenti e caratteri speciali di molte lingue. La tabella 2.2 mostra tutti i tipi di accento applicati alla lettera 'o'. Naturalmente, funzionano anche con altre lettere.

Ai primordi di L<sup>A</sup>T<sub>E</sub>X, per mettere un accento sopra una i o una j, i loro punti dovevano essere rimossi. Questo si ottiene digitando \i e \j. Oggi molti accenti che si usano con la i e la j non richiedono più di ricorrere a questi comandi, se non nei rari casi in cui ciò non succede.

H\^otel, na\"ive, \'el\'eve,\  
 sm\o rrebr\o d, ¡Se\norita!,\  
 Sch\"onbrunner Schlo\ss, Stra\ss e

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

Ma con la codifica UTF-8, e *tastiera permettendo*, si può scrivere direttamente:

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

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

## 2.5 Supporto linguistico internazionale<sup>6</sup>

Quando si scrivono documenti in lingue diverse dall'inglese, ci sono tre aree in cui L<sup>A</sup>T<sub>E</sub>X deve essere configurato in modo appropriato:

<sup>5</sup>Si osservi che in portoghese le legature sopra descritte sono slegate. Al contrario in alfabeti oggi poco usati esistono certe legature storiche rare da incontrare in pubblicazioni moderne. Persino la lettera ß, ancora usata in tedesco, è la legatura di una 's' mediale con una 's' finale. La 'e commerciale', ℰ, è la congiunzione latina *Et* con la 'E' e la 't' legate.

<sup>6</sup>Questa sezione originariamente è stata composta da Axel Kielhorn.

Tabella 2.2: Accenti e caratteri speciali

ò	\‘o	ó	\’o	ô	\^o	õ	\~o
ō	\=o	ò	\.o	ö	\"o	ç	\c{c}
ö	\u{o}	ö	\v{o}	ő	\H{o}	ø	\c{o}
ø	\d{o}	ø	\b{o}	öo	\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’	ı	ı’

1. Tutte le stringhe di testo generate automaticamente<sup>7</sup> vanno adattate alla nuova lingua.
2.  $\LaTeX$  ha bisogno di conoscere le regole di cesura della lingua corrente.
3. Vanno rispettate le regole tipografiche specifiche della lingua corrente. Per esempio, in francese prima di ogni segno di interpunzione, eccetto il punto e la virgola, ci deve essere uno spazio insecabile.

Anche l’inserimento di testo nella lingua scelta potrebbe essere un po’ macchinoso adoperando tutti i comandi mostrati nella tabella 2.2. Per superare questo problema, fino al primo decennio di questo secolo era necessario ricorrere a espedienti diversi per scrivere con le lettere accentate e con i simboli speciali. Oggi, con i moderni motori del sistema  $\TeX$ , che usano in modo nativo la codifica UTF-8, questi problemi sono quasi spariti.

Componendo con  $\text{pdf}\LaTeX$ , è necessario ricorrere alla codifica UTF-8 e a font di 256 caratteri diversi per le varie lingue che usano caratteri speciali; per le lingue dell’Europa occidentale la codifica T1 dei font va sempre bene. La gestione delle lingue è fatta con il pacchetto `babel` insieme a moltissimi file che contengono le regole di cesura per molte lingue ufficiali e per alcune lingue regionali (in totale circa un centinaio).

Componendo con  $\text{Xe}\LaTeX$  o con  $\text{Lua}\LaTeX$ , il pacchetto `polyglossia` [18] sostituisce il pacchetto `babel`. Entrambi i pacchetti si occupano dei pattern di sillabazione, delle stringhe di testo generate automaticamente e delle consuetudini tipografiche nei paesi in cui si scrive in quelle lingue. Inoltre, il pacchetto `fontspec` [20] gestisce il caricamento dei font per  $\text{Xe}\LaTeX$  e  $\text{Lua}\LaTeX$ . Il font predefinito è il Latin Modern.

<sup>7</sup>Indice, Elenco delle figure, Capitolo,...

---

```

\usepackage{fontspec}
\defaultfontfeatures{Ligatures=TeX}
\usepackage{polyglossia}
\setmainlanguage{italian}

```

---

Figura 2.1: La gestione dei font e delle lingue sia per Lua $\text{\LaTeX}$  sia per X $\text{\LaTeX}$

### 2.5.1 Uso di babel

Il pacchetto `babel` si carica nel solito modo, e gli si specificano le lingue da usare nel documento con i nomi (inglesi) delle lingue desiderate; per esempio:

```
\usepackage[italian.main, english, french]{babel}
```

Ovviamente, il suffisso `.main` qualifica la lingua principale del documento.

Si cambia lingua usando uno dei seguenti comandi o ambienti:

```

\selectlanguage{<lingua>}
\foreignlanguage{<lingua>}{<testo>}
\foreignlanguage*{<lingua>}{<testo>}
\begin{otherlanguage}{<lingua>}<testo>\end{otherlanguage}
\begin{otherlanguage*}{<lingua>}<testo>\end{otherlanguage*}

```

I comandi e gli ambienti asteriscati cambiano soltanto le regole di cesura per la `<lingua>` specificata, mentre gli altri modificano anche le parole composte automaticamente. `\selectlanguage` è globale e agisce finché non si torna esplicitamente alla lingua principale. Gli altri comandi svolgono la loro funzione solamente sul `<testo>` specificato. Questi comandi sono comuni anche a `polyglossia`.

### 2.5.2 Uso di polyglossia

A seconda del motore  $\text{\TeX}$  in uso, nel preambolo del documento sono necessari comandi leggermente diversi per abilitare correttamente l'elaborazione multilingue. Il codice 2.1 nella pagina 28 mostra un esempio di parte di preambolo che gestisce font e lingue per i due motori Unicode.

Finora non c'è stato nessuno svantaggio nell'adoperare pdf $\text{\LaTeX}$ . Le cose cambiano quando si abbandona la scrittura latina e si passa a una lingua straniera come il greco o il russo. Con un sistema basato su Unicode, si possono inserire in modo relativamente semplice, *tastiera permettendo*, i caratteri nativi nell'editor e  $\text{\TeX}$  li comprenderà.

Scrivere in lingue diverse è facile, basta specificarle nel preambolo. Questo esempio adopera il pacchetto `csquotes`, che genera il giusto tipo di virgolette

in base alla lingua in cui si scrive. Si noti che deve essere caricato *prima* di caricare il supporto alla lingua.

```
\usepackage[autostyle=true]{csquotes}
\setmainlanguage{italian}
\setotherlanguage{german}
```

Per scrivere un capoverso in tedesco, si può adoperare l'ambiente `german`:

```
Testo italiano.
\begin{german}
Deutscher \enquote{Text}.
\end{german}
Altro \enquote{testo} italiano.
```

```
Testo italiano. Deutscher „Text“. Altro
«testo» italiano.
```

Se bisogna comporre in una lingua straniera una sola parola, o una breve locuzione, si può usare il comando `\langle lingua \rangle text` il cui nome è ottenuto agglutinando quello della `\langle lingua \rangle`<sup>8</sup> con la parola `text`:

```
Sapete che
\germantext{Gesundheit} è
davvero una parola tedesca?
```

```
Sapete che Gesundheit è davvero una
parola tedesca?
```

Questo può sembrare non necessario, poiché l'unico vantaggio è una corretta sillabazione, ma quando la seconda lingua è un po' più esotica, ne varrà la pena.

A volte il font adoperato nel documento principale non contiene i glifi richiesti nella seconda lingua. Il Latin Modern, per esempio, non contiene lettere cirilliche. La soluzione è definire un font che verrà adoperato per quella lingua. Ogni volta che viene attivata una nuova lingua, `polyglossia` controlla prima se un font è stato definito per quella lingua. Se si è soddisfatti del font Computer Modern, si consiglia di provare il font «Computer Modern Unicode» aggiungendo i seguenti comandi al preambolo del documento.

La collezione di font OpenType «Computer Modern Unicode» è piuttosto completa, e come gli altri font OpenType dovrebbe potersi specificare attraverso i comandi specifici di `fontspec` con

```
\setmainfont{CMU Serif}
\setsansfont{CMU Sans Serif}
\setmonofont{CMU Typewriter Text}
```

che qui si sono mostrati per indicare come si farebbe in generale con gli altri font.

Ma questi particolari font «Computer Modern Unicode» richiedono di essere più espliciti, e tanto per `XYLaTeX` quanto per `LuaLaTeX` si deve ricorrere ad un codice più dettagliato:

<sup>8</sup>La `\langle lingua \rangle` va specificata col nome inglese usato da `polyglossia`.

```
\setmainfont{cmun}[Extension=.otf,UprightFont=*rm,
  ItalicFont=*ti,BoldFont=*bx, BoldItalicFont=*bi]
\setsansfont{cmun}[Extension=.otf,UprightFont=*ss,
  ItalicFont=*si,BoldFont=*sx,BoldItalicFont=*so]
\setmonofont{cmun}[Extension=.otf,UprightFont=*btl,
  ItalicFont=*bto,BoldFont=*tb,BoldItalicFont=*tx]
```

Questo particolare testo è composto con pdf $\LaTeX$ ; con i font giusti e la definizione dei comandi `\russiantxt` e `\greetxt` si ottiene lo stesso risultato che, invece, con X $\TeX$  e Lua $\TeX$  si può ottenere direttamente.

La `\russiantxt{Правда}` è un giornale russo. `\greetxt{ἀλήθεια}` in filosofia significa ‘verità’ o ‘scoperta’.

La Правда è un giornale russo.  
La parola ἀλήθεια in filosofia significa ‘verità’ o ‘scoperta’.

## 2.6 Lo spazio tra le parole

Per ottenere un margine destro corretto nell’output,  $\LaTeX$  inserisce spazi di ampiezza variabile tra le parole, leggermente in più alla fine di una frase, in quanto la cosa renderebbe il testo più leggibile.  $\LaTeX$  presuppone che le frasi terminino con punti, punti interrogativi o punti esclamativi. Se un punto segue una lettera maiuscola, non viene preso come un punto di fine frase, poiché i punti dopo le lettere maiuscole si trovano di solito nelle abbreviazioni.

Qualsiasi eccezione a questi presupposti va specificata dall’autore. Una barra rovescia di fronte a uno spazio genera uno spazio che non verrà ingrandito. Un carattere di tilde ‘~’ genera uno spazio che non può essere ingrandito e inoltre vieta un’interruzione di riga. Il comando `\@` prima di un punto specifica che questo punto termina una frase anche se immediatamente prima c’è una lettera maiuscola.

Il sig.~Rossi era felice di vederla.  
Cf.~fig.~5\  
Mi piace il FORTRAN\@. E a te?

Il sig. Rossi era felice di vederla. Cf. fig. 5  
Mi piace il FORTRAN. E a te?

Lo spazio aggiuntivo dopo i segni di interpunzione di fine enunciato può essere disabilitato con il comando

`\frenchspacing`

che dice a  $\LaTeX$  di *non* inserire dopo un segno di fine frase più spazio di quanto ne mette dopo un normale carattere. La cosa è molto comune nelle lingue diverse dall’inglese, tranne che nelle bibliografie. Se si adopera `\frenchspacing`, il comando `\@` non serve.

## 2.7 Titoli, capitoli e paragrafi

Per aiutare il lettore a trovare la propria strada nel suo lavoro, l'autore dovrebbe suddividerlo in capitoli, paragrafi e sottoparagrafi.  $\text{\LaTeX}$  aiuta a farlo con comandi speciali che prendono come argomento il titolo della sezione. Spetta all'autore adoperarli nell'ordine corretto.

I seguenti comandi di sezionamento sono disponibili nelle classi `book` e `report`; invece, nella classe `article` il comando `\chapter` manca:

```

\part{\titolo}
\chapter{\titolo}
\section{\titolo}
\subsection{\titolo}
\subsubsection{\titolo}
\paragraph{\titolo}
\subparagraph{\titolo}
\appendix

```

Siccome la classe `article` non accetta capitoli, è abbastanza facile aggiungere articoli a un libro come se fossero capitoli.

La spaziatura tra le sezioni, la numerazione e il corpo del font dei titoli saranno impostati automaticamente da  $\text{\LaTeX}$ .

Due comandi di sezionamento sono un po' speciali:

- Il comando `\part` non influenza la numerazione dei capitoli e delle altre sezioni.
- Il comando `\appendix` non prende argomenti. Semplicemente, trasforma in lettere la numerazione dei capitoli ai quali cambia anche la parola fissa “Chapter”, “Capitolo”,... in “Appendix”, “Appendice”,... Nella classe `article` modifica la numerazione dei paragrafi.

$\text{\LaTeX}$  crea l'indice generale prendendo i titoli delle sezioni e i numeri delle pagine dall'ultimo ciclo di compilazione del documento. I comandi

```

\tableofcontents   \listoffigures   \listoftables

```

li espandono in un indice generale e negli elenchi delle figure e delle tabelle nel punto in cui vengono dati. Ogni documento deve essere compilato due volte dopo ogni modifica per ottenere l'indice e gli elenchi corretti. Talvolta potrebbe essere necessario compilarlo una terza volta.  $\text{\LaTeX}$  dirà quando serve farlo.

Tutti i comandi di sezionamento sopra elencati esistono anche come versioni “asteriscate”. La versione “asteriscata” di tali comandi viene creata aggiungendo un asterisco `*` dopo i loro nomi. Questo genera titoli di

sezione che non compaiono nell'indice e non sono numerati. Il comando `\section{Aiuto}`, per esempio, potrebbe diventare `\section*{Aiuto}`.

Di solito, i titoli di sezione vengono visualizzati nell'indice esattamente come sono stati immessi nel testo. A volte questo non è possibile, perché il titolo è troppo lungo per stare nell'indice. La voce per l'indice può quindi essere specificata come argomento facoltativo prima del titolo corrente.

```
\chapter[Titolo breve per l'indice]{Un titolo lungo e soprattutto noioso scritto all'inizio del capitolo}
```

Il titolo dell'intero documento viene generato dando il comando

```
\maketitle
```

dopo `\begin{document}` ma prima di qualsiasi altro comando che produca qualunque cosa nel file di uscita.

Il contenuto del titolo va definito prima di dare `\maketitle` (anche nel preambolo) mediante i comandi

```
\title{...}, \author{...} e, facoltativamente, \date{...}
```

Se non interessa la data, bisogna specificarla come `\date{}` altrimenti viene scritta la data del giorno corrente. Nell'argomento di `\author` si possono mettere diversi nomi separati da altrettanti comandi `\and`.

**Attenzione:** Il nome va sempre scritto prima del cognome!

Si può trovare un esempio di alcuni dei comandi menzionati sopra nella figura 1.2 nella pagina 8.

Oltre ai comandi di sezionamento spiegati sopra,  $\text{\LaTeX} 2_{\epsilon}$  ha introdotto tre comandi/dichiarazioni aggiuntivi da adoperare con la classe `book`. Sono utili per suddividere il documento. I comandi alterano le intestazioni dei capitoli e la numerazione delle pagine per funzionare come ci si aspetterebbe in un libro:

`\frontmatter` dovrebbe essere il primissimo comando dopo l'inizio del corpo del documento (`\begin{document}`). La numerazione delle pagine romana invece che in cifre arabe e le sezioni non verranno numerate, come se si stessero adoperando i comandi di sezionamento asteriscati ma le sezioni verranno comunque visualizzate nell'indice. L'indice generale, specialmente se molto articolato, e gli eventuali elenchi delle figure e delle tabelle vanno in questa prima parte del libro.

`\mainmatter` si trova immediatamente prima del primo capitolo del libro. Imposta la numerazione araba delle pagine e riavvia il contatore delle pagine. Tutti i comandi di sezionamento verranno numerati, almeno fino al livello specificato dalla classe, di solito fino al livello dei paragrafi, talvolta fino al livello dei sottoparagrafi.

`\appendix` segna l’inizio di materiale aggiuntivo nel libro. Dopo questo comando, i capitoli verranno numerati con lettere. Questo però non funziona se il comando `\appendix` viene dato dopo `\backmatter`, perché in quest’ultima macrosezione nessun elemento viene numerato.

`\backmatter` dovrebbe essere messo prima degli ultimissimi elementi nel libro, come la bibliografia e l’indice analitico. Le sezioni non vengono numerate e i capitoli possono iniziare in pagine di destra o di sinistra, senza che vengano lasciate pagine bianche.

## 2.8 Riferimenti incrociati

Nei libri, nelle relazioni e negli articoli, spesso si trovano riferimenti incrociati a figure, tabelle e segmenti speciali di testo.  $\LaTeX$  fornisce i seguenti comandi per i riferimenti incrociati:

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

dove *etichetta* è un identificatore univoco scelto dall’utente.  $\LaTeX$  sostituisce `\ref` con il numero di capitolo, o paragrafo, o sottoparagrafo, o figura, o tabella, o equazione, o teorema dopo i quali è stato dato il corrispondente comando `\label`. `\pageref` stampa il numero della pagina dove si trova il comando `\label`.<sup>9</sup> Come per i titoli di sezione e i numeri di pagina per l’indice, vengono adoperati i numeri del precedente ciclo di compilazione.

Un riferimento a questa sezione `\label{sec:this}` assomiglia a:  
 ‘‘vedi il paragrafo-`\ref{sec:this}`  
 nella pagina-`\pageref{sec:this}`.’’

Un riferimento a questa sezione assomiglia  
 a: “vedi il paragrafo 2.8 nella pagina 33.”

## 2.9 Note a piè di pagina

Con il comando

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

si stampa una nota al fondo della pagina corrente. Le note a fondo pagina dovrebbero essere specificate sempre *dopo* la parola o la frase a cui si riferiscono. Le note che si riferiscono a un’intera frase o a una parte di essa dovrebbero perciò essere messe dopo la virgola o il punto.<sup>10</sup>

<sup>9</sup>Si noti che questi comandi non sanno a che cosa si riferiscono; `\label` semplicemente salva l’ultimo numero generato automaticamente.

<sup>10</sup>Si noti che le note distraggono il lettore. Dopo tutto, tutti leggono le note—siamo una specie curiosa, quindi perché non integrare semplicemente nel corpo del documento tutto ciò che si vuole dire?

Le note a fondo pagina `\footnote{Questa ne è un esempio.}` sono adoperate spesso dagli utenti di `\LaTeX`.

Le note a fondo pagina<sup>a</sup> sono adoperate spesso dagli utenti di `LATEX`.

<sup>a</sup>Questa ne è un esempio.

## 2.10 Parole evidenziate

Se un testo è scritto adoperando una macchina per scrivere, le parole importanti sono evidenziate sottolineandole.

```
\underline{<text>}
```

Nei libri a stampa, tuttavia, le parole sono evidenziate componendole con un font *corsivo*. L'autore non dovrebbe preoccuparsene in nessun modo. Il punto importante è dire a `LATEX` che un particolare testo è importante e che deve essere evidenziato. Di qui il comando

```
\emph{<text>}
```

per evidenziare il testo. Ciò che il comando davvero fa con il proprio argomento dipende dal contesto:

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

*Se si evidenzia qualcosa all'interno di un testo già evidenziato `LATEX` usa il font normale per farlo.*

Se si vuole controllare il font e il corpo del font, la sezione 6.2 nella pagina 121 potrebbe fornire qualche ispirazione.

## 2.11 Ambienti

Per comporre testo un po' speciale, `LATEX` dispone di diversi ambienti per realizzare ogni tipo di composizione:

```
\begin{<ambiente> <testo> \end{<ambiente>}
```

Dove `<ambiente>` è il nome dell'ambiente. Gli ambienti possono essere annidati l'uno nell'altro a condizione di mantenere il corretto ordine di annidamento.

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

Nei paragrafi seguenti, vengono spiegati tutti gli ambienti importanti.

### 2.11.1 Liste: `itemize`, `enumerate` e `description`

L'ambiente `itemize` è adatto per elencazioni semplici, l'ambiente `enumerate` per enumerazioni e l'ambiente `description` per liste di definizioni.

```
{\raggedright
  \begin{enumerate}
  \item Si possono annidare gli
    ambienti di questo tipo a piacere:
    \begin{itemize}
      \item ma finirebbe per
        diventare assurdo;
      \item[-] e puoi anche marcare
        con un trattino.
    \end{itemize}
  \item Perciò ricorda:
    \begin{description}
      \item[Scemo] Le cose non
        diventano furbe perché sono
        in un elenco.
      \item[Furbo] Tuttavia, le cose
        possono essere presentate
        elegantemente in un elenco.
    \end{description}
  \end{enumerate}
}
```

- |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> <li>1. Si possono annidare gli ambienti di questo tipo a piacere:           <ul style="list-style-type: none"> <li>• ma finirebbe per diventare assurdo;</li> <li>- e puoi anche marcare con un trattino.</li> </ul> </li> <li>2. Perciò ricorda:           <p><b>Scemo</b> Le cose non diventano furbe perché sono in un elenco.</p> <p><b>Furbo</b> Tuttavia, le cose possono essere presentate elegantemente in un elenco.</p> </li> </ol> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 2.11.2 Testi non giustificati: `flushleft`, `flushright` e `center`

Nell'esempio precedente si è usata la dichiarazione `\raggedright`; le analoghe dichiarazioni `\raggedleft` e `\centering` fanno rispettivamente le stesse cose degli ambienti seguenti. Gli ambienti `flushleft` e `flushright` producono capoversi allineati a sinistra o allineati a destra. L'ambiente `center` produce testo centrato. Se non si dà `\` per specificare le interruzioni di riga, `LATEX` le determinerà automaticamente.

Gli ambienti svolgono la loro funzionalità solo sul loro contenuto; le dichiarazioni mantengono la loro impostazione dentro un gruppo o un ambiente dove sono annidate, purché il gruppo o l'ambiente esplicitamente o implicitamente termini con un "fine capoverso". Gli ambienti mettono ulteriormente in risalto il loro contenuto facendolo precedere e seguire da spazi verticali che talvolta disturbano la composizione. Per esempio, dentro un ambiente `figure` è conveniente usare `\centering` se si vuole centra l'immagine nella giustezza della pagina; se invece si usasse l'ambiente `center`, la figura sarebbe circondata da troppo spazio vuoto.

```
\begin{flushleft}
Questo testo è allineato a sinistra.\\
\LaTeX\ non cerca di comporre le
righe della stessa lunghezza.
\end{flushleft}
```

Questo testo è allineato a sinistra.  
 $\LaTeX$  non cerca di comporre le righe  
della stessa lunghezza.

```
\begin{flushright}
Questo testo è allineato a destra.\\
\LaTeX\ non cerca di comporre le
righe della stessa lunghezza.
\end{flushright}
```

Questo testo è allineato a destra.  
 $\LaTeX$  non cerca di comporre le righe  
della stessa lunghezza.

```
\begin{center}
Questo testo è centrato\\
e le righe sono di lunghezza
diversa.
\end{center}
```

Questo testo è centrato  
e le righe sono di lunghezza diversa.

### 2.11.3 Testi in evidenza: quote, quotation e verse

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

Una regola tipografica a spanne per stabilire la giustezza è la seguente:

```
\begin{quote}
In media nessuna riga di testo
dovrebbe contenere più di 66
caratteri.
\end{quote}
```

Questo è il motivo per il quale le pagine composte con  $\LaTeX$  hanno margini così ampi, e anche perché nei giornali si compone su molte colonne.

Una regola tipografica a spanne per stabilire la giustezza è la seguente:

In media nessuna riga di testo dovrebbe contenere più di 66 caratteri.

Questo è il motivo per il quale le pagine composte con  $\LaTeX$  hanno margini così ampi, e anche perché nei giornali si compone su molte colonne.

Esistono due ambienti simili: gli ambienti `quotation` e `verse`. L'ambiente `quotation` è utile per citazioni più lunghe formate da alcuni capoversi, poiché ne rientra la prima riga. L'ambiente `verse` è utile per le poesie, in cui le interruzioni delle righe sono importanti. I versi sono separati tra loro mediante un `\\` alla fine di ciascuno, e una riga vuota dopo ogni strofa. Nell'esempio che segue l'ambiente `verse` è annidato dentro un ambiente `flushleft`; è conveniente farlo perché i versi di una poesia, anche se sono formati tutti con lo stesso numero di sillabe, hanno lunghezze diverse e se devono andare a capo, con il rientro, come si vede nell'esempio, è bene che lo facciano senza

introdurre cesure. Nell'esempio che segue la giustezza è così corta che tutti i versi vanno a capo; potendo disporre dell'intera giustezza del blocco del testo, è difficile che si presenti la necessità di ricorrere esplicitamente alla composizione in bandiera.

```
L'unica poesia inglese che conosco
a memoria è quella di Humpty Dumpty.
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\
Humpty Dumpty had a great fall.\
All the King's horses and all
the King's men\
Couldn't put Humpty together
again.
\end{verse}
\end{flushleft}
```

L'unica poesia inglese che conosco a memoria è quella di Humpty Dumpty.

Humpty Dumpty sat on a  
wall:  
Humpty Dumpty had a great  
fall.  
All the King's horses and all  
the King's men  
Couldn't put Humpty  
together again.

#### 2.11.4 L'estratto

Nelle pubblicazioni scientifiche, è consuetudine incominciare con un estratto, o breve sommario, che offre al lettore una rapida panoramica di cosa aspettarsi. Per questo scopo  $\text{\LaTeX}$  con le classi `report` e `article` fornisce l'ambiente `abstract`.

```
\begin{abstract}
L'estratto astratto.
\end{abstract}
```

#### Sommario

L'estratto astratto.

#### 2.11.5 Stampare alla lettera

Il testo racchiuso tra `\begin{verbatim}` e `\end{verbatim}` verrà stampato direttamente, come se fosse scritto con una macchina per scrivere, con tutte le interruzioni di riga e gli spazi, senza che alcun comando  $\text{\LaTeX}$  venga eseguito.

All'interno di un capoverso, è possibile ottenere un comportamento simile con

```
\verb+text+ 
```

Il `+` è solo un esempio di carattere delimitatore. Si adoperi qualunque carattere diverso da una lettera, per esempio, `*` o `|`. Molti esempi  $\text{\LaTeX}$  in questa guida sono composti con questo comando.

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

```
\begin{verbatim}
10 PRINT "HELLO WORLD";
20 GOTO 10
\end{verbatim}
```

Il comando `\ldots...`

```
10 PRINT "HELLO WORLD";
20 GOTO 10
```

Il comando `\verb` può essere adoperato in modo simile con un asterisco, e lo spazio viene reso visibile con questo segno `␣`:

`\verb*|come in questo caso:-) |`

```
\begin{verbatim*}
La versione asteriscata
dell'ambiente verbatim evidenzia
gli spazi nel testo da comporre
alla lettera.
\end{verbatim*}
```

`come␣in␣questo␣caso:-)␣`

```
La␣versione␣asteriscata
dell'ambiente␣verbatim␣evidenzia
gli␣spazi␣nel␣testo␣da␣comporre
alla␣lettera.
```

L'ambiente `verbatim` e il comando `\verb` non possono essere adoperati nell'argomento di altri comandi.

### 2.11.6 Tabelle

L'ambiente `tabular` può essere adoperato per comporre belle tabelle con filetti orizzontali e verticali facoltativi.  $\text{\LaTeX}$  determina automaticamente la larghezza delle colonne.

L'argomento *specifiche* del comando

```
\begin{tabular}[\langle posizione \rangle]{\langle specifiche \rangle}
```

definisce il formato della tabella. Si adopera `l` per una colonna di celle di testo allineato a sinistra, `r` per testo allineato a destra e `c` per testo centrato; `p{\langle width \rangle}` per una colonna di celle contenenti testo giustificato e con interruzioni di riga; `|` per un filetto verticale.

Se il testo in una colonna di tipo `l`, `c`, e `r` è troppo largo perché la tabella intera resti nella giustezza della la pagina,  $\text{\LaTeX}$  non lo ripiegherà su più righe. Adoperando `p{\langle larghezza \rangle}`, invece, si definisce uno speciale tipo di colonna che conterrà celle il cui testo è un capoverso normale con giustezza pari a `\langle larghezza \rangle`.

L'argomento *posizione* specifica la posizione verticale della tabella rispetto alla linea di base del testo circostante.<sup>11</sup> Si adopera una delle lettere `t`, `b` e `c` per specificare l'allineamento della tabella in alto, in basso e al centro rispettivamente.

<sup>11</sup>In realtà, solo con il comando di allineamento `c`, allinea l'asse matematico del testo circostante con l'asse matematico della tabella.

In un ambiente `tabular`, `&` passa alla colonna successiva, `\\` incomincia una nuova riga e `\hline` inserisce un filetto orizzontale. Si inseriscono filetti orizzontali parziali adoperando `\cline{<i-j>}`, dove  $i$  e  $j$  sono i numeri delle colonne iniziale e finale fra cui dovrebbe estendersi il filetto. I filetti verticali si inseriscono con `|`, oppure con `\vline` di cui si dirà fra poco.

Si ottengono risultati migliori con il pacchetto `booktabs`, che è raccomandato per comporre tabelle molto professionali, quasi prive di filetti orizzontali e assolutamente prive di filetti verticali.

```
\begin{tabular}{|r|l|}
\hline
7C0 & esadecimale \\
3700 & ottale \\
\cline{2-2}
11111000000 & binario \\
\hline
1984 & decimale \\
\hline
\end{tabular}
```

7C0	esadecimale
3700	ottale
11111000000	binario
1984	decimale

L'ambiente `tabular` può anche essere usato per incorniciare un piccolo medaglione con un intero (breve) capoverso.

```
\begin{tabular}{|p{55mm}|}
\hline
Benvenuti nel capoverso
riquadrato. Spero che vi
piaccia.\\
\hline
\end{tabular}
```

Benvenuti nel capoverso riquadrato. Spero che vi piaccia.
--------------------------------------------------------------

Il separatore di colonna può essere specificato con il costrutto `@{...}`. Questo comando elimina lo spazio intercolonna predefinito e lo sostituisce con qualunque cosa si trovi tra le due parentesi graffe. Un uso comune di questo comando è spiegato di seguito a proposito del problema dell'allineamento decimale. Un'altra possibile applicazione è quella di eliminare lo spazio iniziale in una tabella con `@{}`. Negli esempi che seguono, la presenza o l'assenza degli spazi a sinistra delle prima colonna e a destra dell'ultima colonna sono evidenziati dai filetti orizzontali. I filetti verticali non si dovrebbero mai usare; ma se si vogliono inserire nel costrutto `@{...}`, invece di `|` si deve usare `\vline`.

```
\begin{tabular}{@{} l @{} }
\hline
Senza spazi esterni\\
\hline
\end{tabular}
```

<u>Senza spazi esterni</u>
----------------------------

```
\begin{tabular}{l}
\hline
Con spazi esterni a destra
e a sinistra\\
\hline
\end{tabular}
```

Con spazi esterni a destra e a sinistra
-----------------------------------------

```
\begin{tabular}{@{\vline}l@{\vline}}
\hline
Con filetti verticali
adiacenti al contenuto\\
\hline
\end{tabular}
```

Con filetti verticali adiacenti al contenuto
----------------------------------------------

### Incolonnamento del separatore decimale

Mike Ressler

Poiché non esiste un modo nativo per allineare le colonne numeriche sulla base di un separatore decimale,<sup>12</sup> si può ricorrere a un “inganno” e ottenere il risultato adoperando due colonne: la parte intera allineata a destra e quella decimale allineata a sinistra. Il comando `@{,}` nell’argomento della riga `\begin{tabular}` sostituisce il normale spazio tra le colonne con una semplice virgola, dando l’apparenza di una sola colonna giustificata sulla base della virgola decimale. Non bisogna dimenticare di sostituire il separatore decimale nei numeri con un separatore di colonna (`&`). Sopra la nostra “colonna” numerica si può mettere un’intestazione di colonna adoperando il comando `\multicolumn`.

```
\begin{tabular}{cr@{,}l}
Potenze di $e$ & & \\
\multicolumn{2}{c}{Valore} \\
\hline
$e$ & & 2&7183 \\
$e^e$ & & 15&154 \\
$e^{e^e}$ & & 1618&2 \\
\end{tabular}
```

Potenze di e	Valore
e	2,7183
e <sup>e</sup>	15,154
e <sup>e<sup>e</sup></sup>	1618,2

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Domenica} \\
\hline
Lunedì & Sabato \\
\hline
Martedì & Venerdì \\
\hline
Mercoledì & Giovedì \\
\hline
\end{tabular}
```

Domenica	
Lunedì	Sabato
Martedì	Venerdì
Mercoledì	Giovedì

<sup>12</sup>Si vedano però i pacchetti `dcolumn` e `siunitx`.

Il materiale composto con l'ambiente `tabular` rimane sempre in un'unica pagina. Se si vogliono comporre tabelle lunghe, si può adoperare l'ambiente `longtable` fornito dal pacchetto `longtable`.

A volte, le tabelle  $\text{\LaTeX}$  predefinite sembrano un po' sacrificate. Quindi si può dare loro un po' più di respiro impostando un valore più alto di `\arraystretch` e di `\tabcolsep`. Viceversa, si può riportare dentro la larghezza delle specchio di stampa una tabella un po' troppo larga; questo effetto si ottiene diminuendo un poco il valore di `\tabcolsep`. Nell'esempio che segue, la prima tabella è composta con le impostazioni di default; nella seconda tabella lo stiramento delle righe fornito dall'argomento del comando `\arraystretch` è rivelato dalla maggior distanza dei filetti orizzontali; invece l'aumento dello spazio intercolonna,, specificato con l'argomento di `\tabcolsep` visto che esiste una sola colonna, si vede dallo spazio maggiore tra i filetti verticali e il contenuto della colonna.

```
\begin{tabular}{|l|}
\hline
Queste righe \ \ \hline
sono strette \ \ \hline
\end{tabular}\ \ \ [3ex]
{\renewcommand{\arraystretch}{1.5}
\setlength{\tabcolsep}{5mm}
\begin{tabular}{|l|}
\hline
Queste righe hanno\ \ \hline
più respiro \ \ \ \ \ \hline
\end{tabular}}
```

Queste righe sono strette
Queste righe hanno più respiro

Se si vuole solo aumentare l'altezza di una singola riga nella tabella, si aggiunga una barra verticale invisibile<sup>13</sup>. Si adopera un `\rule` di larghezza nulla per implementare questo trucco.

```
\begin{tabular}{|l|}
\hline
\rule[-1ex]{1pt}{4ex}%
col pilastrino visibile \ \
\hline
\rule[-1ex]{0pt}{4ex}%
col pilastrino invisibile\ \
\hline
\end{tabular}
```

col pilastrino visibile
col pilastrino invisibile

I simboli `pt` e `ex` nell'esempio precedente sono unità di misura di  $\text{\TeX}$ . Maggiori informazioni sulle unità sono riportate nella tabella 6.5 nella pagina 128.

<sup>13</sup>Nella composizione tipografica tradizionale si chiama 'pilastrino'.

## 2.12 Includere figure e immagini

Come si è spiegato nella sezione precedente, L<sup>A</sup>T<sub>E</sub>X fornisce le funzioni per lavorare con oggetti flottanti, come figure o tabelle, con gli ambienti `figure` e `table`.

Un buon insieme di comandi per includere immagini in questi oggetti flottanti è fornito con il pacchetto `graphicx`. Esso fa parte di un'intera famiglia di pacchetti denominata "graphics bundle".<sup>14</sup>

Per includere un'immagine nel documento, si segua la seguente procedura passo dopo passo:

1. I file contenenti immagini possono essere gestiti con L<sup>A</sup>T<sub>E</sub>X solo se hanno i formati EPS, PDF, PNG o JPEG. Se si crea l'immagine con un programma grafico, la si salvi in uno di questi formati.
2. Quando la figura, nei formati prescritti, viene scaricata da una macchina fotografica digitale o da un altro dispositivo, o dalla rete, potrebbe essere necessario tagliare parti bianche al contorno; talvolta conviene tagliare via dall'immagine parti non interessanti. Quando l'immagine è ottenuta tramite uno scanner, può essere necessario non solo tagliare via contorni o parti superflue, ma potrebbe anche essere indispensabile ruotarla. Tutte queste operazioni si possono fare con uno dei vari editor grafici, tra i quali si può suggerire il programma libero GIMP (GNU Image Manipulation Program) che però non mantiene la vettorialità dei formati EPS e PDF; per questi formati si possono usare sia Ghostscript o la sua interfaccia grafica GView; per le piattaforme Mac il programma Anteprema.app (Preview.app se il Sistema Operativo è in inglese) permette di fare queste operazioni senza tuttavia essere pesante con funzionalità non necessarie.
3. Si carichi il pacchetto `graphicx` nel preambolo del file principale del documento da comporre:

```
\usepackage{graphicx}
```

4. Si dia il comando

```
\includegraphics[chiave = valore,...]{nome-del-file}
```

per includere il *file* nel documento. L'argomento facoltativo accetta una lista di *chiavi* e relativi *valori* separate con la virgola. Le *chiavi* possono essere adoperate per modificare larghezza, altezza e orientamento dell'immagine inclusa. La tabella 2.3 nella pagina 43 elenca le chiavi più importanti.

---

<sup>14</sup>CTAN://pkg/graphics

Tabella 2.3: Alcune chiavi per il pacchetto `graphicx`

<code>width</code>	scala l'immagine alla larghezza specificata
<code>height</code>	scala l'immagine all'altezza specificata
<code>angle</code>	ruota l'immagine dell'angolo specificato
<code>scale</code>	scala l'immagine con un solo fattore di scala
<code>keepaspectratio</code>	sceglie un solo fattore di scala in orizzontale e in verticale in modo da non deformare l'immagine

Il codice 2.1 nella pagina 43 può aiutare a chiarire le cose. Non occorre specificare l'estensione del file immagine; nell'esempio, se esiste nella cartella di lavoro un file immagine di nome `test.pdf`, o `test.eps`, o `test.jpg` o `test.png`, il comando `\includegraphics` include la figura memorizzata nel file che ha trovato. Bisogna specificare l'estensione solo se nella cartella di lavoro ci sono più file con lo stesso nome ma con una diversa estensione.

Nell'esempio, la figura viene *prima* ruotata di un angolo di  $90^\circ$  e *poi* ridimensionata alla larghezza finale di 0,5 volte la larghezza di un capoverso standard. Il rapporto altezza/base è mantenuto uguale a quello dell'immagine inclusa, perché non si è specificata alcuna altezza particolare. I parametri di larghezza e altezza possono anche essere specificati in dimensioni assolute invece che proporzionali alle dimensioni della gabbia del testo. Si faccia riferimento alla tabella 6.5 nella pagina 128 per ulteriori informazioni. Se si vuole saperne di più su quest'argomento, si legga la documentazione [9].

## 2.13 Oggetti flottanti

Oggi, la maggior parte delle pubblicazioni contiene molte figure e tabelle. Questi elementi richiedono un trattamento speciale, perché non possono essere suddivisi tra le pagine. Un metodo sarebbe quello di iniziare una nuova pagina ogni volta che una figura o una tabella è troppo grande per adattarsi allo spazio disponibile nella pagina corrente. Questo approccio lascerebbe le pagine parzialmente vuote, e questo è un difetto tipografico molto negativo.

La soluzione a questo problema è far flottare qualunque figura o tabella che non stia nella pagina corrente in una pagina successiva, mentre si continua a riempire la pagina corrente con il corpo del testo.  $\text{\LaTeX}$  offre due ambienti per oggetti flottanti; uno per le tabelle e uno per le figure. Per trarre il massimo

---

```
\includegraphics[angle=90,width=0.5\textwidth]{test}
```

Codice 2.1: Codice d'esempio per includere il file `test` in un documento.

Tabella 2.4: Autorizzazioni di posizionamento

Specificatore	Autorizzazione di posizionamento
<b>h</b>	( <i>here</i> ) possibilmente qui
<b>t</b>	( <i>top</i> ) in testa alla pagina
<b>b</b>	( <i>bottom</i> ) in basso nella pagina
<b>p</b>	( <i>page</i> ) in una pagina contenente solo oggetti flottanti
<b>!</b>	trascurando la maggior parte dei parametri, <sup>a</sup> che potrebbero impedire il posizionamento.

<sup>a</sup>Come per esempio la frazione massima della pagina riservata ai float in posizione **t**.

vantaggio da questi due ambienti, è importante capire approssimativamente in che modo il programma gestisce internamente oggetti come questi. Altrimenti, essi possono diventare una delle principali fonti di frustrazione, perché  $\text{\LaTeX}$  non li mette mai dove si vorrebbe che stessero.

Diamo prima un'occhiata ai comandi che disponibili per gli oggetti flottanti.

```

\begin{figure}[\langle specificatore di posizionamento \rangle]
oppure
\begin{table}[\langle specificatore di posizionamento \rangle]
```

Qualunque materiale si trovi racchiuso in un ambiente **figure** o **table** verrà trattato come materiale flottante. Entrambi gli ambienti flottanti accettano un parametro facoltativo, chiamato  $\langle specificatore di posizionamento \rangle$ . Questo parametro viene adoperato per comunicare a  $\text{\LaTeX}$  le posizioni in cui è desiderabile spostare l'oggetto. Uno  $\langle specificatore di posizionamento \rangle$  è costruito assemblando una stringa di *autorizzazioni di posizionamento flottante*. Si veda la tabella 2.4.

Per esempio, una tabella flottante potrebbe essere inserita specificando la seguente linea di codice:

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

Lo specificatore di posizionamento **[\!hbp]** permette a  $\text{\LaTeX}$  di mettere la tabella possibilmente qui (**h**) o nella parte bassa (**b**) di qualche pagina o in una pagina speciale di soli oggetti flottanti (**p**), e tutto questo anche se non sembra così bello (**!**). Se non si specifica alcun  $\langle posizionamento \rangle$ , le classi standard assumono come stringa predefinita **[tbp]**.

$\text{\LaTeX}$  posizionerà ogni oggetto flottante che incontra secondo lo specificatore di posizionamento fornito dall'autore. Se un oggetto non può essere

messo sulla pagina corrente, viene rimandato alla coda delle *figure* o a quella delle *tabelle*.<sup>15</sup> Quando viene incominciata una nuova pagina, L<sup>A</sup>T<sub>E</sub>X controlla prima se è possibile riempire una speciale pagina con gli oggetti flottanti presenti nelle code. Se questo non è possibile, il primo oggetto flottante di ciascuna coda viene trattato come se lo si fosse appena incontrato nel testo: L<sup>A</sup>T<sub>E</sub>X prova nuovamente a posizionarlo in base ai rispettivi specificatori di posizione (eccetto ‘h’, perché non è più possibile).

Qualsiasi nuovo oggetto mobile presente nel testo viene inserito nelle code appropriate. L<sup>A</sup>T<sub>E</sub>X mantiene rigorosamente l’ordine di apparizione originale per ogni tipo di oggetto flottante. Ecco perché una figura che non può essere posizionata spinge tutte le ulteriori figure alla fine del documento. Perciò:

Se L<sup>A</sup>T<sub>E</sub>X non sta posizionando gli oggetti flottanti come ci si aspetterebbe, spesso è perché uno solo di essi blocca una delle code.

Anche se è possibile fornire a L<sup>A</sup>T<sub>E</sub>X gli specificatori di posizionamento per una singola posizione, ciò causa problemi. Se l’oggetto flottante non ci sta nella posizione specificata, si blocca, bloccando quelli successivi. In particolare, non bisogna mai e poi mai adoperare l’opzione [h] da sola; è così brutto che nelle versioni più recenti di L<sup>A</sup>T<sub>E</sub>X viene automaticamente sostituito da [ht].

Dopo aver spiegato la parte difficile, ci sono alcune altre cose da menzionare sugli ambienti `table` e `figure`. Il comando

```
\caption{testo della didascalia}
```

serve per definire una didascalia per l’oggetto flottante. Verranno aggiunti da L<sup>A</sup>T<sub>E</sub>X la parola “Figura” o “Tabella” seguita da un numero progressivo.

I due comandi

```
\listoffigures    \listoftables
```

agiscono analogamente al comando `\tableofcontents`, stampando un elenco di figure e, rispettivamente, uno di tabelle. Queste liste mostreranno l’intera didascalia, perciò se si ha la tendenza ad adoperare didascalie lunghe, bisogna averne una versione più breve per gli elenchi. Questo si ottiene inserendo la versione breve tra parentesi quadre dopo il comando `\caption`.

```
\caption[⟨didascalia breve⟩]{⟨didascalia lunga⟩}
```

Si usi `\label` e `\ref` per creare un riferimento a un oggetto flottante nel testo. Si noti che il comando `\label{⟨etichetta⟩}` va dato *dopo* il comando `\caption`, dato che il suo `⟨etichetta⟩` si deve riferire al numero della didascalia.

---

<sup>15</sup>Esse sono code del tipo FIFO — *First In, First Out* cioè «il primo che entra è il primo che esce».

L'esempio seguente disegna un quadrato e lo inserisce nel documento. Lo si può adoperare se si volesse riservare spazio per le immagini che verranno incollate nel documento finito e stampato su carta.

La figura~\ref{white} è un esempio di Pop-Art.

```
\begin{figure}[!hbt]
\includegraphics[angle=90,width=\textwidth]{white-box.pdf}
\caption{Scatola bianca di Peter Markus Paulian}
\label{white}
\end{figure}
```

Nell'esempio qui sopra, L<sup>A</sup>T<sub>E</sub>X tenterà in tutti i modi di mettere la figura proprio *qui* (**h**).<sup>16</sup> Se la cosa non è possibile, cerca di metterla *in alto* (**t**) o in basso (**b**) nella pagina (indipendentemente dall'ordine col quale sono scritti i posizionatori **t** e **b**). Non riuscendo a posizionare la figura sulla pagina corrente, determina se è possibile creare una pagina di soli oggetti flottanti contenente questa figura e forse alcune tabelle della coda delle tabelle. Se non c'è abbastanza materiale per una pagina speciale di oggetti flottanti, L<sup>A</sup>T<sub>E</sub>X incomincia una nuova pagina e ancora una volta tratta la figura come se l'avesse appena incontrata nel testo.

In determinate circostanze, potrebbe essere necessario adoperare i comandi

<code>\clearpage</code>	<code>\cleardoublepage</code>
-------------------------	-------------------------------

Essi ordinano a L<sup>A</sup>T<sub>E</sub>X di svuotare immediatamente le code degli oggetti flottanti ancora in coda e, successivamente, di incominciare una nuova pagina; `\cleardoublepage`, inoltre, incomincia una nuova pagina di destra.

---

<sup>16</sup>Assumendo che la coda delle figure sia vuota.

## Capitolo 3

# Comporre matematica

Ora tutto è pronto! In questo capitolo cominciamo a trattare il punto di maggior forza di  $\text{T}_{\text{E}}\text{X}$ : la matematica. Ma attenzione: questo capitolo resta solo in superficie. Mentre le cose qui descritte sono sufficienti per molte persone, non bisogna disperarsi se non si riesce a trovare in queste pagine la soluzione per certi particolari problemi di composizione matematica. Molto probabilmente il problema è risolto dal pacchetto `amsmath`.

### 3.1 Il pacchetto `amsmath`

Se si vuole comporre matematica (avanzata), bisogna usare le funzionalità messe a disposizione dall'American Mathematical Society in una collezione di pacchetti e file di classe. Noi ci riferiremo principalmente al pacchetto `amsmath` che, insieme ad altri pacchetti installati, costituisce una collezione estremamente utile per comporre matematica non elementare.  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  da solo fornisce molti strumenti di base per comporre la matematica; l'aggiunta del pacchetto `amsmath` rende  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  di una potenza estrema.

I pacchetti della collezione `amsmath` sono già presenti in ogni installazione completa ed aggiornata del sistema  $\text{T}_{\text{E}}\text{X}$ .

In questo capitolo assumiamo che il pacchetto `amsmath` sia già stato caricato avendo inserito nel preambolo la direttiva

```
\usepackage{amsmath}
```

### 3.2 Equazioni isolate

Una formula matematica può essere composta in linea con il testo di un capoverso, oppure si può interrompere il capoverso per inserire la formula in stile *fuori testo*, per poi riprenderne la composizione dopo la formula. Le

formule matematiche in linea col testo vengono immesse racchiudendole fra due segni \$.

Aggiungi il quadrato di  $a$  al quadrato di  $b$  per ottenere il quadrato di  $c$ . Oppure, usando la scrittura matematica:  
 $a^2 + b^2 = c^2$

Aggiungi il quadrato di  $a$  al quadrato di  $b$  per ottenere il quadrato di  $c$ . Oppure, usando la scrittura matematica:  $a^2 + b^2 = c^2$

$\text{\TeX}$  si pronuncia  
 $\tau\epsilon\chi$   
 100 m<sup>3</sup> d'acqua  
 Questo viene dal mio ♡

$\text{\TeX}$  si pronuncia  $\tau\epsilon\chi$   
 100 m<sup>3</sup> d'acqua  
 Questo viene dal mio ♡

Se si vuole che le equazioni siano composte separate dal resto di un capoverso, è preferibile comporre *fuori testo* (in *display*, come dice in inglese) piuttosto che dividere il capoverso in due distinti capoversi. Per fare questo si racchiude la formula tra le direttive  $\text{\begin{equation}}$  e  $\text{\end{equation}}$ .<sup>1</sup> A questo punto è possibile etichettare l'equazione con il comando  $\text{\label}$  e far riferimento al numero dell'equazione con il comando  $\text{\eqref}$ . Si può anche assegnare un nome all'equazione, invece che un numero, usando il comando  $\text{\tag}$ .

Aggiungi il quadrato di  $a$  al quadrato di  $b$  per ottenere il quadrato di  $c$ . In termini matematici:  
 $\text{\begin{equation}}$   
 $a^2 + b^2 = c^2$   
 $\text{\end{equation}}$

Aggiungi il quadrato di  $a$  al quadrato di  $b$  per ottenere il quadrato di  $c$ . In termini matematici:

$$a^2 + b^2 = c^2 \quad (3.1)$$

Einstein afferma che:  
 $\text{\begin{equation}}$   
 $E = mc^2$   $\text{\label{giusto}}$   
 $\text{\end{equation}}$

Einstein afferma che:

$$E = mc^2 \quad (3.2)$$

<sup>1</sup>Il pacchetto `amsmath` fornisce anche l'ambiente `equation*` che non numera l'equazione. Se per qualche oscuro motivo non si vogliono usare le funzionalità del pacchetto `amsmath`, si può usare al suo posto l'ambiente nativo del nucleo di  $\text{\LaTeX}$  `displaymath`, che però ha prestazioni limitate.

Non ha detto che:  
`\begin{equation*}`  
`1 + 1 = 3 \tag{errato}`  
`\end{equation*}` e questo è un  
 riferimento all'equazione  
`\eqref{giusto}`.

Non ha detto che:

$$1 + 1 = 3 \quad (\text{errato})$$

e questo è un riferimento all'equazione  
 (3.2).

Se non si vuole che L<sup>A</sup>T<sub>E</sub>X numeri le equazioni, si usi la versione asteriscata `equation*` del pacchetto `amsmath` oppure, ancora più semplicemente, fra i delimitatori `\[` e `\]`:<sup>2</sup>.

Aggiungi il quadrato di  $a$  al quadrato di  $b$  per ottenere il quadrato di  $c$ ; in matematica:  
`\begin{equation*}`  
`a^2 + b^2 = c^2`  
`\end{equation*}`  
 oppure puoi ottenere lo stesso risultato scrivendo di meno:  
`\[ a^2 + b^2 = c^2 \]`

Aggiungi il quadrato di  $a$  al quadrato di  $b$  per ottenere il quadrato di  $c$ ; in matematica:

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

oppure puoi ottenere lo stesso risultato scrivendo di meno:

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

Mentre `\[...]` è più corto, non permette di passare facilmente da una espressione numerata ad una non numerata, e viceversa, come `equation` e `equation*`.

Si noti la differenza nello stile di composizione delle equazioni fra lo stile testo e lo stile fuori testo:

Questo è lo stile testo:  
`$$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$`  
 E questo è lo stile fuori testo:  
`\begin{equation}`  
`\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2}`  
`= \frac{\pi^2}{6}`  
`\end{equation}`

Questo è lo stile testo:  $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$ . E questo è lo stile fuori testo:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (3.3)$$

In modo testo L<sup>A</sup>T<sub>E</sub>X usa uno stile più raccolto in senso verticale così da non alterare la spaziatura delle righe; se però è necessario inserire in modo testo espressioni matematiche un po' alte e/o profonde e, per qualche motivo, non le si vuole comporre fuori testo, si può tentare di usare il comando

<sup>2</sup>. Come già annotato, il nucleo di L<sup>A</sup>T<sub>E</sub>X ha solo l'ambiente `equation` senza la versione asteriscata.

`\smash`; esso non modifica il suo argomento, ma ne nasconde l'altezza e la profondità. L'inconveniente è che le parti più alte o più profonde si possono sovrapporre a elementi della riga precedente o successiva. Quindi non è detto che l'uso di `\smash` sia risolutivo.

Un'espressione matematica `\$d_{e_p}\$` seguita da un'altra espressione `\$h^{i^g^h}\$`.  
A confronto con un'espressione schiacciata `\smash{\$d_{e_p}\$}` con un'altra espressione schiacciata `\smash{\$h^{i^g^h}\$}`.

Un'espressione matematica  $d_{e_p}$  seguita da un'altra espressione  $h^{i^g^h}$ . A confronto con un'espressione schiacciata  $d_{e_p}$  con un'altra espressione schiacciata  $h^{i^g^h}$ .

### 3.2.1 Il modo matematico

Ci sono anche altre differenze fra la composizione fuori testo e quella in linea col testo.

1. La maggior parte degli spazi e dei fine riga non hanno nessun significato poiché gli spazi sono dedotti logicamente dall'espressione stessa, oppure sono stati specificati espressamente mediante comandi speciali come `\,`, `\quad` o `\qquad` (torneremo su questo punto più avanti nel paragrafo 3.7).
2. Non sono ammesse righe vuote. È come se la formula formasse un solo capoverso.
3. Ogni lettera è considerata come il nome di una variabile e verrà composta come tale. Se si vuole comporre del testo normale all'interno di una formula (con font normale e spaziatura normale) bisogna introdurlo come argomento del comando `\text` del pacchetto `amsmath` (vedi anche il paragrafo 3.9 nella pagina 65).

`\forall x \in \mathbf{R} :`  
`\quad \quad x^2 \geq 0`

$\forall x \in \mathbf{R} : \quad x^2 \geq 0$

`x^2 \geq 0 \quad \quad \quad`  
`\text{per ogni } x \in \mathbf{R}`

$x^2 \geq 0 \quad \text{per ogni } x \in \mathbf{R}$

I matematici sono molto pignoli in merito ai simboli da usare: in questo esempio sarebbe necessario usare il font *blackboard bold* (il neretto da lavagna, dove, scrivendo col gesso, il neretto è rappresentato raddoppiando i tratti verticali delle lettere) che si ottiene con il comando `\mathbb` del pacchetto `amsmath` richiamato a sua volta dal pacchetto `amssymb` (cioè se si carica

amssymn ci pensa questo a caricare amsfonts). In questo modo l'ultimo esempio diventa:

```
$x^{2} \geq 0 \quad \text{per ogni } x \in \mathbb{R}
```

$$x^2 \geq 0 \quad \text{per ogni } x \in \mathbb{R}$$

Si veda la tabella 3.15 nella pagina 75 per esaminare alcuni font matematici.

### 3.3 Come si costruisce una formula matematica

In questo paragrafo descriviamo i comandi più importanti da usare per comporre la matematica. La maggior parte di essi non richiede le funzionalità del pacchetto amsmath (in caso contrario, lo specificheremmo chiaramente), ma lo si carichi lo stesso.

Le **lettere greche minuscole** si inseriscono con i loro nomi (inglesi) per disteso: `\alpha`, `\beta`, `\gamma`, ...; le lettere maiuscole sono introdotte nello stesso modo: `\Gamma`, `\Delta`, ...<sup>3</sup>

```
\lambda, \xi, \pi, \theta, \mu, \Phi, \Omega, \Delta
```

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

Gli **esponenti**, gli **apici**, i **pedici** vengono specificati come argomenti dei caratteri speciali `^` e `_`. La maggior parte dei comandi agisce solo sul carattere successivo, per cui se si vuole estendere l'azione su una stringa di caratteri, la si deve racchiudere fra parentesi graffe: `{...}`.

La tabella 3.4 nella pagina 72 elenca molti simboli di relazione come  $\subseteq$  e  $\perp$ .

```
$p^3_{ij} \quad m_{Knuth} \quad \sum_{k=1}^3 k  
a^x + y \neq a^{x+y} \quad e^{x^2} \neq e^{x^2}
```

$$p_{ij}^3 \quad m_{Knuth} \quad \sum_{k=1}^3 k$$

$$a^x + y \neq a^{x+y} \quad e^{x^2} \neq e^{x^2}$$

La **radice quadrata** viene immessa con il comando `\sqrt`; la radice  $n$ -esima è generata con `\sqrt[n]`. Le dimensioni verticali e orizzontali della radice sono determinate automaticamente da L<sup>A</sup>T<sub>E</sub>X. In alcune nazioni è normale usare il solo segno senza l'argomento né il filetto orizzontale, usando il comando `\surd`, ma non lo si faccia: è vietato dalle norme ISO.

I vari tipi di frecce come  $\leftrightarrow$  e  $\Rightarrow$  sono mostrati nella tabella 3.7 nella pagina 73.

<sup>3</sup>Non sono definite in L<sup>A</sup>T<sub>E</sub>X quelle maiuscole greche come Alpha, Beta, e alcune altre che sono identiche alle lettere latine.

`\sqrt{x}`  $\Leftrightarrow x^{1/2}$   
`\quad \sqrt[3]{2}`  
`\quad \sqrt{x^2 + \sqrt{y}}`  
 ma non si scriva mai una cosa come  
`\[\surd[x^2 + y^2]\]` perché è vietato  
 dalle norme ISO.

$\sqrt{x} \Leftrightarrow x^{1/2}$     $\sqrt[3]{2}$     $\sqrt{x^2 + \sqrt{y}}$   
 ma non si scriva mai una cosa come

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

perché è vietato dalle norme ISO.

Mentre il segno di **punto** per indicare la moltiplicazione generalmente viene omesso, talvolta può essere utile esplicitarlo e lo si fa con il comando `\cdot` che inserisce un singolo punto a metà altezza. Invece `\cdots` inserisce tre puntini a metà altezza e `\ldots` inserisce tre puntini sulla linea di base. Oltre a questi puntini se ne possono mettere tre in verticale con `\vdots` o in diagonale con `\ddots`. Ci sono altri esempi nel paragrafo 3.6.

`\Psi = v_1 \cdot v_2`  
`\cdot \ldots \quad`  
`n! = 1 \cdot 2`  
`\dots (n-1) \cdot n`

$$\Psi = v_1 \cdot v_2 \dots \quad n! = 1 \cdot 2 \dots (n-1) \cdot n$$

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

`0.\overline{3} =`  
`\underline{\underline{1/3}}`

$$0.\overline{3} = \underline{\underline{1/3}}$$

I comandi `\overbrace` e `\underbrace` creano delle graffe orizzontali lunghe quanto basta per raggruppare sopra o sotto gli elementi di una espressione.

`\underbrace{\overbrace{a+b+c}^6`  
`\cdot \overbrace{d+e+f}^7`  
`}_{\text{età della vita}} = 42`

$$\underbrace{\overbrace{a+b+c}^6 \cdot \overbrace{d+e+f}^7}_{\text{età della vita}} = 42$$

Per aggiungere alle variabili accenti matematici come **piccole frecce** o **tildi larghe**, si usano i comandi esposti nella tabella 3.2 nella pagina 71. I circonflessi larghi e le tildi larghe possono coprire diversi caratteri e si impostano con `\widetilde` e `\widehat`. Si noti la differenza fra `\hat` e `\widehat` e la posizione di `\bar` per una variabile con un indice. L'apostrofo ' dà il segno di primo; due distinti apostrofi danno il segno di secondo.

`f(x) = x^2 \quad f'(x)`  
`= 2x \quad f''(x) = 2`  
`\hat{XY} \quad \widehat{XY}`  
`\quad \bar{x}_0 \quad \bar{\bar{x}}_0`

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

$$\hat{XY} \quad \widehat{XY} \quad \bar{x}_0 \quad \bar{\bar{x}}_0$$

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

Tabella 3.1: Comandi per i nomi di alcune funzioni matematiche

I **vettori** sono spesso specificati aggiungendo una piccola freccia sopra la loro variabile. Questo si ottiene con il comando `\vec`. I comandi `\overrightarrow` e `\overleftarrow` sono utili per indicare il vettore tra  $A$  e  $B$ :

```

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

```

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

I nomi delle funzioni sono sempre composti nel font tondo diritto, non come le variabili che vengono composte in corsivo; si può usare il corsivo solo quando la funzione è indicata con una sola lettera e sono esplicitati i suoi argomenti fra parentesi; per le altre funzioni, L<sup>A</sup>T<sub>E</sub>X fornisce i comandi mostrati nella tabella 3.1.

```

Ecco un limite matematico
\begin{equation*}
\lim_{x \to 0}
\frac{\sin x}{x}=1
\end{equation*}
ben noto agli studenti del liceo
scientifico.

```

Ecco un limite matematico

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

ben noto agli studenti del liceo scientifico.

Per definire comandi per comporre i nomi di altre funzioni, oltre a quelli elencati nella tabella 3.1, si usi il comando `\DeclareMathOperator` di `amsmath`. Esiste anche la versione asteriscata per definire i nomi di operatori con limiti. Questi comandi funzionano solo nel preambolo perciò le linee commentate nell'esempio che segue devono essere messe nel preambolo del documento (ovviamente senza il segno di commento).

```

% \DeclareMathOperator{\argh}{argh}
% \DeclareMathOperator*{\nut}{Nut}
Due operatori strani
\begin{equation*}
3\argh_0(x,1) = 2\nut_{x=1}
\end{equation*}
il cui significato è misterioso.

```

Due operatori strani

$$3\argh_0(x,1) = 2\nut_{x=1}$$

il cui significato è misterioso.

Per l'aritmetica modulare ci sono due comandi: `\bmod` per l'operatore binario “ $a \bmod b$ ”, e `\pmod`:

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

$$a \bmod b$$

$$x \equiv a \pmod{b}$$

Una *frazione* viene composta usando il comando `\frac` con la sintassi

$$\frac{\langle \text{numeratore} \rangle}{\langle \text{denominatore} \rangle}$$

Nelle espressioni in linea col testo la frazione viene rimpicciolita per restare nella riga. Questo stile di composizione può essere ottenuto anche nelle frazioni fuori testo usando il comando `\tfrac`. Al contrario, lo stile di composizione fuori testo può essere ottenuto anche in linea col testo se si usa il comando `\dfrac`. Spesso si usa la frazione con la barra, come  $1/2$ , che ha un aspetto migliore se numeratore e denominatore sono semplici.

```
In modalità fuori testo:  
\begin{equation*}  
3/8 \quad \frac{3}{8}  
\quad \tfrac{3}{8}  
\end{equation*}
```

In modalità fuori testo:

$$3/8 \quad \frac{3}{8} \quad \frac{3}{8}$$

```
In modalità in linea:  
$1\frac{1}{2}$-ore,  
$1\dfrac{1}{2}$-ore.
```

In modalità in linea:  $1\frac{1}{2}$  ore,  $1\frac{1}{2}$  ore.

Nell'esempio che segue, viene usato il comando `\partial` per indicare la derivata parziale:

```
\begin{equation*}  
\sqrt{\frac{x^2}{k+1}} \quad \frac{\partial^2 f}{\partial x^2}  
x^{\frac{2}{k+1}}  
\end{equation*}
```

$$\sqrt{\frac{x^2}{k+1}} \quad x^{\frac{2}{k+1}} \quad \frac{\partial^2 f}{\partial x^2}$$

Per comporre il coefficiente binomiale, o strutture simili, si usa il comando `\binomial` di `amsmath`

```
La regola di Pascal è:  
\begin{equation*}  
\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}  
\end{equation*}
```

La regola di Pascal è:

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

Per le relazioni binarie può essere necessario usare simboli formati da due segni impilati l'uno sull'altro. Il comando `\stackrel{apice}{base}`, appunto, impila l'apice sopra la base; nel far questo compone il primo argomento col corpo degli apici.

L'espressione

```
\begin{equation*}
f_n(x)
\stackrel{\mathrm{def}}{\geq} 1
\end{equation*}
```

Serve per ricordare che per definizione la funzione è sempre maggiore o uguale a 1.

L'espressione

$$f_n(x) \stackrel{\text{def}}{\geq} 1$$

Serve per ricordare che per definizione la funzione è sempre maggiore o uguale a 1.

Il *segno di integrale* è generato con il comando `\int`, il *segno di somma* è generato con il comando `\sum`, e il *segno di prodotto* è generato con il comando `\prod`.

```
\begin{equation*}
\sum_{i=1}^n \quad \int_0^{\frac{\pi}{2}} \quad \prod_{\epsilon}
\end{equation*}
```

$$\sum_{i=1}^n \quad \int_0^{\frac{\pi}{2}} \quad \prod_{\epsilon}$$

Il pacchetto `amsmath` fornisce anche i comandi per gli integrali multipli; vedi più avanti.

Per controllare meglio la posizione degli indici in espressioni complesse, il pacchetto `amsmath` fornisce anche il comando `\substack`:

```
\begin{equation*}
\sum_{\substack{0 < i < n \\ j \subseteq i}} P(i, j) = Q(i, j)
\end{equation*}
```

$$\sum_{\substack{0 < i < n \\ j \subseteq i}} P(i, j) = Q(i, j)$$

$\LaTeX$  fornisce anche ogni tipo di delimitatori per racchiudere sotto-espressioni di formule complesse; molti di essi sono anche estensibili verticalmente o orizzontalmente in relazione al loro contenuto. Le parentesi tonde e quadre possono essere immesse con i tasti della propria tastiera; le parentesi graffe vanno immesse con i comandi `\{` e `\}`; tutti gli altri delimitatori sono generati da comandi specifici (per esempio `\updownarrow`).

```
\begin{equation*}
\{a, b, c\} \neq \{a, b, c\}
\end{equation*}
```

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

Se si mettono `\left`, prima di un delimitatore di apertura, e `\right`, prima di un delimitatore di chiusura,  $\LaTeX$  provvede a scalare i due delimi-



Il modo più comodo per andare a capo è quello di servirsi dell'ambiente `multline` del pacchetto `amsmath`.

```
\begin{multline}
a + b + c + d + e + f
+ g + h + i
\\
= j + k + l + m + n
\end{multline}
```

$$\begin{aligned}
 a + b + c + d + e + f + g + h + i \\
 = j + k + l + m + n \quad (3.4)
 \end{aligned}$$

La differenza rispetto all'ambiente `equation` è che si può inserire un numero qualsiasi di comandi di “a capo” inserendo il comando `\\` nei punti in cui è lecito spezzare la formula. Similmente a `equation*`, esiste anche l'ambiente `multline*`; entrambi permettono di comporre una formula senza numerarla.

Si consideri la situazione seguente:

```
\begin{equation}
a = b + c + d + e + f
+ g + h + i + j
+ k + l + m + n + o + p
\label{eq:equation_too_long}
\end{equation}
```

$$a = b + c + d + e + f + g + h + i + j + k + l + m + n + o + p \quad (3.5)$$

Qui, in effetti, è il secondo membro la parte più lunga che non sta nella giustezza. L'ambiente `multline` produce il risultato seguente:

```
\begin{multline}
a = b + c + d + e + f
+ g + h + i + j \\
+ k + l + m + n + o + p
\end{multline}
```

$$\begin{aligned}
 a = b + c + d + e + f + g + h + i + j \\
 + k + l + m + n + o + p \quad (3.6)
 \end{aligned}$$

Questo è meglio dell'equazione (3.5) ma ha il difetto che il segno di uguaglianza perde la sua rilevanza rispetto al segno di somma che precede  $k$ .

## 3.5 Insiemi di equazioni

Nel caso più generale, disponiamo di diverse equazioni che non stanno nella giustezza. Qui dobbiamo lavorare con un allineamento verticale in modo da mantenere l'insieme di equazioni in una struttura elegante e leggibile.

Prima di procedere, esaminiamo alcuni brutti esempi che mostrano i più comuni inconvenienti che si incontrano con l'ambiente `eqnarray` del nucleo di  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .

### 3.5.1 Problemi con i comandi tradizionali

Per raggruppare un insieme di equazioni, si può usare l'ambiente `eqnarray`<sup>4</sup> nativo di  $\text{\LaTeX}$ :

```
\begin{eqnarray}
a &=& b + c \\
&& d + e
\end{eqnarray}
```

$$a = b + c \quad (3.7)$$

$$= d + e \quad (3.8)$$

Questo ambiente (criticabile per i motivi che vedremo fra poco) fallisce non appena una linea è troppo lunga della dimensione giusta.<sup>5</sup>

```
\begin{eqnarray}
a &=& b + c \\
&& d + e + f + g + h + i \\
&& + j + k + l \nonumber \\
&& + m + n + o \\
&& p + q + r + s
\end{eqnarray}
```

$$a = b + c \quad (3.9)$$

$$= d + e + f + g + h + i + j + k + l \\ + m + n + o \quad (3.10)$$

$$= p + q + r + s \quad (3.11)$$

Qui il termine  $+m$  non dovrebbe essere incolonnato sotto  $d$  e nemmeno sotto il segno di uguaglianza, ma con  $+m$  incolonnato sotto  $+e$ . In effetti basterebbe inserire fra il segno `&` e il termine  $+m$  semplicemente `\quad`, ma, come in questo esempio, è meglio regolare con finezza lo spazio usando il comando `\mkern` specifico per le spaziature matematiche (vedi tab. 6.5).

```
\begin{eqnarray}
a &=& & b + c \\
&=& & d + e + f + g + h + i \\
&=& & + j + k + l \nonumber \\
&=& & + m + n + o \\
&=& & p + q + r + s
\end{eqnarray}
```

$$a = b + c \quad (3.12)$$

$$= d + e + f + g + h + i + j + k + l \\ + m + n + o \quad (3.13)$$

$$= p + q + r + s \quad (3.14)$$

Questo è il momento in cui confrontare l'ambiente `eqnarray` del nucleo di  $\text{\LaTeX}$  con l'ambiente `align` del pacchetto `amsmath`.

Infatti, usando `eqnarray` gli spazi attorno al segno di uguaglianza sono troppo grandi. In particolare essi non sono larghi esattamente come quelli degli ambienti `multline` e `equation`:

<sup>4</sup>L'ambiente `eqnarray` può essere usato per raggruppare diversi blocchi di equazioni

<sup>5</sup>Come dice il suo nome, il comando `\nonumber` impedisce la numerazione di una riga dell'incolonnamento creato con l'ambiente `eqnarray`.

Come si vede

```
\begin{eqnarray}
a & = & a = a
\end{eqnarray}
```

gli spazi attorno al segno di uguaglianza sono diversi

Come si vede

$$a = a = a \quad (3.15)$$

gli spazi attorno al segno di uguaglianza sono diversi

Inoltre, l'espressione talvolta si sovrappone al numero della formula anche se c'è abbastanza spazio a sinistra.

```
\begin{eqnarray}
a & = & b + c \\
& = & d + e + f + g + h^2 \\
& & + i^2 + j
\label{eq:faultryeqnarray}
\end{eqnarray}
```

$$\begin{aligned} a &= b + c & (3.16) \\ &= d + e + f + g + h^2 + i^2 + j & (3.17) \end{aligned}$$

L'ambiente `eqnarray` di  $\text{\LaTeX}$  presenta anche altri difetti che si manifestano meno frequentemente, ma più gravi, come quello di essere incompatibile con il simbolo di “fine dimostrazione” nell'ambito dell'ambiente `proof`.

Però esso offre il comando `\lefteqn` che si può usare quando il primo membro è troppo lungo.

Insieme di espressioni che si spingono troppo a destra

```
\begin{eqnarray}
\lefteqn{a + b + c + d
+ e + f + g + h}\nonumber \\
& = & i + j + k + l + m \\
& & \\
& = & n + o + p + q + r + s
\end{eqnarray}
```

Insieme di espressioni che si spingono troppo a destra

$$\begin{aligned} a + b + c + d + e + f + g + h \\ &= i + j + k + l + m & (3.18) \\ &= n + o + p + q + r + s & (3.19) \end{aligned}$$

Anche quest'altra composizione non è ottimale, perché il secondo membro è troppo corto e non risulta ben centrato.

Composizione non ottimale:

```
\begin{eqnarray}
\lefteqn{a + b + c + d
+ e + f + g + h} \\
\nonumber \\
& = & i + j
\end{eqnarray}
```

Composizione non ottimale:

$$\begin{aligned} a + b + c + d + e + f + g + h \\ &= i + j & (3.20) \end{aligned}$$

### 3.5.2 L'ambiente `align`

Avendo caricato il pacchetto `amsmath`, come si è raccomandato di fare, il problemi che `eqnarray` presenta possono essere risolti con l'ambiente `align`, uno dei molti ambienti di allineamento verticale delle equazioni introdotti da `amsmath`.

Intanto lo spazio attorno al segno di uguale è delle dimensioni giuste:

```
\begin{align}
a &= b + c \\
&= d + e + f + g + h + i \\
&+ j + k + l \\
&= m + n + o + p
\end{align}
```

$$a = b + c \quad (3.21)$$

$$= d + e + f + g + h + i + j + k + l \quad (3.22)$$

$$= m + n + o + p \quad (3.23)$$

Se un'equazione numerata è troppo lunga, `align` evita che possa sovrapporsi al suo numero:

```
\begin{align}
a &= b + c \\
&= d + e + f + g + h^2 \\
&+ i^2 + j
\end{align}
```

$$a = b + c \quad (3.24)$$

$$= d + e + f + g + h^2 + i^2 + j \quad (3.25)$$

Se il primo membro è troppo lungo, il pacchetto `amsmath` offre un secondo ambiente `aligned` che permette di incolonnare e di spezzare parti di un incolonnamento:

```
\begin{align}
\begin{aligned}[b]
a + b + c + d \\
{} + e + f + g + h
\end{aligned} \\
&= i + j + k + l \quad \backslash\text{nonumber}\backslash \\
&= m + n + o + p + q + r
\end{align}
```

$$a + b + c + d$$

$$+ e + f + g + h = i + j + k + l$$

$$= m + n + o + p + q + r \quad (3.26)$$

Il risultato, come si vede, non è sgradevole. Si noti, piuttosto, che l'argomento facoltativo `[b]` (b: *bottom*) assegnato al comando di apertura dell'ambiente `aligned` garantisce che la sua ultima riga sia allineata con il successivo segno di uguaglianza.

Tra gli ambienti di `amsmath` compare anche l'ambiente `subequations`, che permette di numerare le equazioni di un sistema con una numerazione "letterale", subordinata all'unica numerazione dell'intero sistema, a cui si

può fare riferimento: (3.27), come lo si può fare a una qualunque particolare equazione del sistema: (3.27c):

```
\begin{subequations}
  \label{система}
  \begin{align}
    \nabla\times\vec{E}&=
    -\frac{\partial\vec{B}}{\partial t}\backslash
    \nabla\times\vec{H}&=
    \frac{\partial\vec{D}}{\partial t}+\vec{J}\backslash
    \nabla\cdot\vec{D}&= \rho
    \label{система-3}\backslash
    \nabla\cdot\vec{B}&= 0
  \end{align}
\end{subequations}
```

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (3.27a)$$

$$\nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{J} \quad (3.27b)$$

$$\nabla \cdot \vec{D} = \rho \quad (3.27c)$$

$$\nabla \cdot \vec{B} = 0 \quad (3.27d)$$

Come si vede, con gli ambienti del pacchetto `amsmath` si possono risolvere di solito molto bene tutti i problemi che riguardano sia le espressioni che si svolgono su più righe, sia gli insiemi di equazioni incolonnate.

Il pacchetto `amsmath` contiene numerosi altri ambienti per gli incolonnamenti oltre a `align`: `align*`, `gather`, `gather*`, `alignat`, `alignat*`, `falign`, `falign*`, `split`, `aligned`, `gathered`. Si consiglia fortemente al lettore di leggere la documentazione del pacchetto `amsmath` perché è della massima utilità, tanto che dovrebbe essere sempre usato per comporre qualunque documento che contenga anche poca matematica, ma di livello superiore a quella adatta alle scuole elementari.

## 3.6 Matrici

Nel mondo di  $\text{\LaTeX}$  le tabelle con celle composte solo in modo matematico sono chiamate *array* e si compongono con l'ambiente `array`; esso opera nello stesso modo dell'ambiente `tabular` e il comando `\` termina ogni riga. Le tabelle delimitate mediante parentesi di vario genere, o con altri delimitatori, si chiamano *matrici*, *determinanti*, *norme*, e simili, a seconda dei delimitatori usati. Una matrice, scritta usando solo  $\text{\LaTeX}$ , si compone come un `array` delimitato; il pacchetto `amsmath`, come vedremo più avanti, mette a disposizione altri ambienti specifici. L'esempio che segue è una matrice composta con i soli mezzi di  $\text{\LaTeX}$ :

Ecco una matrice delimitata da parentesi tonde

```
\begin{equation*}
\mathbf{X} = \left(
\begin{array}{ccc}
x_1 & x_2 & \ldots \\
x_3 & x_4 & \ldots \\
\vdots & \vdots & \ddots
\end{array}
\right)
\end{equation*}
```

Ecco una matrice delimitata da parentesi tonde

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots \\ x_3 & x_4 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Si noti che quando si usa il comando `\text` in matematica, questo deve contenere all'inizio o alla fine del suo argomento uno spazio, perché in matematica, ma non con `\text`, gli spazi del file input sono totalmente ignorati; la matematica di L<sup>A</sup>T<sub>E</sub>X inserisce gli spazi necessari nelle formule, ma non attorno al testo annidato nelle formule.

L'ambiente `array` può essere anche usato per comporre funzioni discontinue; lo fa usando il solo delimitatore di sinistra, e rimpiazzando il delimitatore di destra con `.` che, come argomento di `\left` o di `\right`, permette di completare la coppia, ma non rappresenta nessun delimitatore.<sup>6</sup>

```
\begin{equation*}
|x| = \left\{
\begin{array}{rl}
-x & \text{if } x < 0, \\
0 & \text{if } x = 0, \\
x & \text{if } x > 0
\end{array}
\right.
\end{equation*}
```

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0 \end{cases}$$

L'ambiente `cases` di `amsmath` rende più semplice il codice, perciò è meglio dargli un'occhiata:

```
\begin{equation*}
|x| =
\begin{cases}
-x & \text{if } x < 0, \\
0 & \text{if } x = 0, \\
x & \text{if } x > 0
\end{cases}
\end{equation*}
```

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0 \end{cases}$$

Si è visto che le matrici possono essere composte con `array`; il pacchetto `amsmath` fornisce una soluzione migliore con sei diversi ambienti per comporre matrici: `matrix` (non delimitata), `pmatrix` (delimitata con parentesi tonde), `bmatrix` (delimitata con parentesi quadre), `Bmatrix` (delimitata con parentesi

<sup>6</sup>I comandi `\left` e `\right` devono *sempre* essere appaiati e richiedono sempre un argomento; il punto rappresenta un carattere che L<sup>A</sup>T<sub>E</sub>X riesce ad elaborare, ma indica l'assenza di un delimitatore visibile. Quanto è contenuto fra `\left` e `\right` serve a L<sup>A</sup>T<sub>E</sub>X per dimensionare correttamente i delimitatori estensibili.

graffe), `vmatrix` (delimitata da barre verticali), `Vmatrix` (delimitata da doppie barre verticali). Non è necessario specificare il numero delle colonne, come succede con `array`. Il massimo numero di colonne è 10, ma lo si può configurare per le proprie esigenze (sebbene sia abbastanza raro che in un documento si debbano comporre matrici con più di 10 colonne).

```
\begin{equation*}
\begin{matrix}
1 & 2 \\
3 & 4
\end{matrix} \quad
\begin{bmatrix}
p_{11} & p_{12} & \dots & p_{1n} \\
p_{21} & p_{22} & \dots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{m1} & p_{m2} & \dots & p_{mn}
\end{bmatrix}
\end{equation*}
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \quad \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \dots & p_{mn} \end{bmatrix}$$

Va notato che le matrici composte con gli ambienti specifici forniti dal pacchetto `amsmath` sono composte in modo più compatto rispetto a quando si compongono con `array`; si confrontino le due matrici del seguente esempio.

```
\begin{equation*}\left(
\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}
\right) \quad
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{pmatrix}
\end{equation*}
```

$$\left( \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array} \right) \quad \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

### 3.7 Le spaziature in modo matematico

Nei rari casi in cui gli spazi inseriti da  $\text{\LaTeX}$  non siano adeguati, si possono aggiustare mediante l'inserimento di comandi speciali: `\`, per uno spazio sottile:  $\frac{3}{18}$  quad = 3 mu ( $\llcorner$ ); `\:` per uno spazio medio:  $\frac{4}{18}$  quad = 4 mu ( $\llcorner$ ); `\;` per uno spazio grande:  $\frac{5}{18}$  quad = 5 mu ( $\llcorner$ ). Il normale comando di spaziatura `\_` produce uno spazio delle dimensioni di quello che appare fra

le parole in modo testo, e `\quad` produce lo spazio di un quadrato ( $\square$ ), mentre `\qqquad` produce lo spazio di un quadratone (due quadrati:  $\square\square$ ). Convenzionalmente si tratta di spazi di `1em` o di `2em`; il nome dell'unità di misura "em", discende dal fatto che in molti font esso è molto prossimo al valore della larghezza della lettera 'M' nel font corrente.<sup>7</sup> Il comando `\!` produce uno spazio negativo piccolo tanto quanto quello positivo prodotto da `\,`, ma in verso opposto:  $-\frac{3}{18}em = -3mu$ .

```
\begin{equation*}
\int_1^2 \ln x \mathrm{d}x
\qqquad
\int_1^2 \ln x \!,\mathrm{d}x
\end{equation*}
```

$$\int_1^2 \ln x dx \quad \int_1^2 \ln x dx$$

Merita notare che il segno di differenziale è spesso composto in tondo diritto. Nel prossimo esempio definiamo un nuovo comando `\ud` (*upright d*) come "d" (si noti lo spazio `\!` prima della d) perciò non dobbiamo riscriverlo ogni volta. La definizione con `\newcommand` va messa nel preambolo.

```
\newcommand{\ud}{\!,\mathrm{d}}

\begin{equation*}
\int_a^b f(x)\ud x
\end{equation*}
```

$$\int_a^b f(x) dx$$

Se si vogliono comporre integrali multipli, si scopre subito che la spaziatura fra i segni di integrale è troppo grande; la si può correggere usando `\!`, ma il pacchetto `amsmath` fornisce una soluzione migliore, cioè definisce direttamente i comandi per comporre i segni giusti per gli integrali doppi, tripli e con molteplicità maggiore: `\iint`, `\iiint` e `\idotsint`.

```
\newcommand{\ud}{\!,\mathrm{d}}

\begin{gather*}
\int\int f(x)g(y) \ud x \ud y \\
\int\!\!\!\!\!\int f(x)g(y) \ud x \ud y \\
\iint f(x)g(y) \ud x \ud y
\end{gather*}
```

$$\int \int f(x)g(y) dx dy$$

$$\int \int f(x)g(y) dx dy$$

$$\int \int f(x)g(y) dx dy$$

Ci si riferisca alla documentazione `amsmath.pdf` o si consulti il testo *The L<sup>A</sup>T<sub>E</sub>X Companion* [3] per maggiori dettagli.

<sup>7</sup>Vedi anche la tabella 6.5.

### 3.8 I caratteri fantasma

Quando si deve allineare verticalmente qualcosa che si trovi ad esponente o a pedice, talvolta è molto utile servirsi di caratteri fantasma. Usando il comando `\phantom` si può riservare spazio per caratteri che non compariranno nel documento composto. Il modo migliore per capire questa operazione è quello di esaminare un esempio:

```
\begin{equation*}
{}^{14}_{6}\text{C}
\quad \text{verso} \quad
{}^{14}_{6}\text{C}
\end{equation*}
```

$${}^{14}_{6}\text{C} \quad \text{verso} \quad {}^{14}_{6}\text{C}$$

Se si vogliono comporre numerosi simboli di isotopi come nell'esempio, consiglieri anche il pacchetto di Niderbergere `mhchem` per comporre oltre agli isotopi anche le formule chimiche in generale.

### 3.9 Giocare con i font matematici

Nella tabella 3.15 nella pagina 75 sono mostrati diversi font matematici.

```
Re \quad
\mathcal{R} \quad
\mathfrak{R} \quad
\mathbb{R} \quad
\mathbb{R}
```

$$\Re \quad \mathcal{R} \quad \mathfrak{R} \quad \mathbb{R}$$

Gli ultimi due font richiedono il pacchetto `amssymb`

Talvolta è necessario indicare a  $\text{\LaTeX}$  il corpo corretto dei font da usare. In modo matematico sono disponibili i comandi seguenti.

```
\displaystyle (123^4),
\textstyle (123^4),
\scriptstyle (123^4) e
\scriptscriptstyle (123^4).
```

Se il simbolo  $\sum$  è usato in una frazione, esso viene composto in stile testo a meno che non si ordini a  $\text{\LaTeX}$  di fare diversamente:

```
\begin{equation*}
P = \frac{\displaystyle
\sum_{i=1}^n (x_i - x)
(y_i - y)}
{\left[
\sum_{i=1}^n (x_i - x)^2
\sum_{i=1}^n (y_i - y)^2
\right]^{1/2}}
\end{equation*}
```

$$P = \frac{\sum_{i=1}^n (x_i - x)(y_i - y)}{\left[ \sum_{i=1}^n (x_i - x)^2 \sum_{i=1}^n (y_i - y)^2 \right]^{1/2}}$$

Cambiare stile generalmente influisce anche su come sono gestiti i grandi operatori e sul modo di comporre apici e pedici.

### 3.9.1 I simboli neri

È abbastanza difficile ottenere simboli in nero con L<sup>A</sup>T<sub>E</sub>X; probabilmente è intenzionale, visto che i tipografi dilettanti tendono a usarli troppo. Il comando per cambiare font, `\mathbf`, produce in effetti lettere nere, ma solo in tondo (diritto) mentre i caratteri matematici sono solitamente in corsivo e inoltre non funziona con le lettere greche minuscole. C'è il comando `\boldmath`, ma *questo può essere usato solo prima di aprire un qualunque ambiente matematico*. Però funziona su tutti i segni matematici, non solo sulle lettere.

```

 $\mu + M$  \quad
 $\mathbf{\mu} + \mathbf{M}$ 
\quad  $\boldmath{\mu + M}$ 

```

$\mu + M$	$\mu + M$	$\mu + M$
-----------	-----------	-----------

Il vecchio pacchetto `amsbsy` è distribuito ancora per retrocompatibilità con la collezione di `amsmath`, ma la documentazione ne sconsiglia l'uso; viene invece consigliato il pacchetto `bm` della collezione `tools`; il compito di avere alcuni simboli in nero, non tutta la formula, è reso molto più semplice visto che fornisce il comando `\bm` (`amsbsy` disponeva del comando `\boldsymbol`):

```

 $\mu + M$  \quad
 $\bm{\mu} + \bm{M}$ 

```

$\mu + M$	$\mu + M$
-----------	-----------

Ci si ricordi che il nero diritto e il nero inclinato hanno in matematica significati diversi e quindi è necessario poter disporre sia del nero diritto e di quello corsivo con tutti i font matematici. La cosa 'è diventata abbastanza comune oggi, anche perché esistono norme specifiche dell'ISO (International Standards Organisation) e riprese dall'UNI (ente di UNificazione Italiana), per cui, almeno nei settori scientifici e tecnologici, esse hanno valore di legge.

## 3.10 Teoremi, lemmi, e altri enunciati

Quando si compone matematica, probabilmente è necessario comporre enunciati come i "lemmi", le "definizioni", gli "assiomi" e altri enunciati di questo genere. Si possono definire con il comando `\newtheorem`.

<code>\newtheorem{&lt;nome&gt;}[&lt;contatore&gt;]{&lt;testo&gt;}[&lt;sezione&gt;]</code>
-------------------------------------------------------------------------------------------

Il `<nome>` è quello di un ambiente per l'enunciato omonimo; il `<testo>` è il titolino che effettivamente viene scritto prima dell'enunciato vero e proprio

e generalmente è numerato con un contatore avente lo stesso  $\langle nome \rangle$ ; se si specifica il  $\langle contatore \rangle$ , che può essere quello di un altro enunciato, viene usato lo stesso contatore, e i due tipi di enunciati vengono numerati in sequenza. Se si specifica il nome  $\langle sezione \rangle$  di un comando di sezionamento, il contatore dell'enunciato viene azzerato (e la numerazione ricomincia poi da 1) quando viene incrementato il contatore di quella sezione; in definitiva, il numero completo dell'enunciato è composto dal numero completo della sezione nel quale l'enunciato si trova, seguito dal numero progressivo dell'enunciato all'interno di quella sezione.

Dopo aver usato il comando `\newtheorem` nel preambolo, l'ambiente dell'enunciato può essere usato così:

```
\begin{ $\langle nome \rangle$ }[ $\langle testo \rangle$ ]
 $\langle testo \text{ dell'enunciato} \rangle$ 
\end{ $\langle nome \rangle$ }
```

Il pacchetto `amsthm` (della collezione di `amsmath`) fornisce il comando `\theoremstyle{ $\langle stile \rangle$ }`, che permette di definire ciò che si riferisce all'enunciato specifico secondo alcuni stili predefiniti `definition` (titolino nero, enunciato in tondo); `plain` (titolino nero, enunciato in corsivo); `remark` (titolino in corsivo, enunciato in tondo).

Questo per la sintassi dovrebbe essere sufficiente. Gli esempi che seguono tolgono ogni dubbio residuo e rendono chiaro il concetto che il comando `\newtheorem` fornisce funzionalità abbastanza complesse.

Per prima cosa si definiscono gli enunciati:

```
\theoremstyle{definition} \newtheorem{law}{Legge}
\theoremstyle{plain}      \newtheorem{jury}[law]{Giuria}
\theoremstyle{remark}    \newtheorem*{margh}{Margherita}
```

Poi si possono usare come in questo esempio.

```
\begin{law} \label{law:box}
Non nasconderti nella scatola
delle prove.
\end{law}
\begin{jury}[I giurati]
Potresti essere tu, quindi stai
attenta e osserva
la legge~\ref{law:box}.
\end{jury}
\begin{jury}
Non osserverai l'ultimo enunciato.
\end{jury}
\begin{margh}No, No, No!\end{margh}
\begin{margh}Gino!\end{margh}
```

**Legge 1.** Non nasconderti nella scatola delle prove.

**Giuria 2** (I giurati). *Potresti essere tu, quindi stai attenta e osserva la legge 1.*

**Giuria 3.** *Non osserverai l'ultimo enunciato.*

*Margherita.* No, No, No!

*Margherita.* Gino!

L'enunciato "Giuria" usa lo stesso contatore dell'enunciato "Legge" perciò riceve un numero in sequenza con quello della Legge. L'argomento facoltativo per il comando di apertura degli ambienti di questo tipo serve per fissare un titolo all'enunciato.

```
\newtheorem{mur}{Murphy}[section]
```

```
\begin{mur}
Se ci sono due o più modi
per fare qualche cosa e
uno di questi modi può produrre
una catastrofe, allora qualcuno
la farà in quel modo.
\end{mur}
```

**Murphy 3.10.1.** Se ci sono due o più modi per fare qualche cosa e uno di questi modi può produrre una catastrofe, allora qualcuno la farà in quel modo.

L'enunciato "Murphy" riceve un numero legato a quello della sezione corrente. Si potrebbe voler usare un altro livello di sezionamento, per esempio il capitolo o, al contrario, il sottoparagrafo.

Se si vogliono personalizzare gli enunciati fino all'ultimo dettaglio, allora il pacchetto `ntheorem` offre una moltitudine di opzioni. Se ne veda la documentazione.

### 3.11 Dimostrazioni e il simbolo di "fine dimostrazione"

Il pacchetto `amsthm` definisce anche l'ambiente `proof`.

```
\begin{proof}
Semplice! Si usa così:
\begin{equation*}
E=mc^2.
\end{equation*}
\end{proof}
\noindent ma con del materiale
fuori testo il simbolo di "fine
dimostrazione" è troppo distanziato.
```

*Dimostrazione.* Semplice! Si usa così:

$$E = mc^2.$$

□

ma con del materiale fuori testo il simbolo di "fine dimostrazione" è troppo distanziato.

Con il comando `\qedhere` si può spostare il simbolo di "fine dimostrazione" dove sta meglio, cioè nella stessa riga dell'equazione.

```
\begin{proof}
Semplice! Si usa così:
\begin{equation*}
E=mc^2. \qedhere
\end{equation*}
\end{proof}
```

`\noindent`  
e in questo modo il simbolo di “fine dimostrazione” non produce troppo spazio vuoto verticale.

*Dimostrazione.* Semplice! Si usa così:

$$E = mc^2. \quad \square$$

e in questo modo il simbolo di “fine dimostrazione” non produce troppo spazio vuoto verticale.

Se una dimostrazione finisce con una o più equazioni in display, come visto sopra, eventualmente numerate, si hanno dei problemi a causa dello spazio che le circonda. Se si usa la versione asteriscata di `alignn` o di uno degli altri ambienti per sistemi di equazioni di `amsmath`, il segno di “fine dimostrazione” viene messo a fine riga in modo normale.

```
\begin{proof}
Questa è una dimostrazione
che finisce con un incolonnamento
di equazioni:
\begin{align*}
a &= b + c \\
&= d + e \qedhere
\end{align*}
\end{proof}
```

*Dimostrazione.* Questa è una dimostrazione che finisce con un incolonnamento di equazioni:

$$\begin{aligned} a &= b + c \\ &= d + e \end{aligned} \quad \square$$

Se invece si vuole che le equazioni siano numerate, il comando `\qedhere` occuperebbe il posto riservato al numero dell’ultima equazione, quindi bisogna prevedere, molto intuitivamente, una riga ulteriore:

```
\begin{proof}
Questa è una dimostrazione
che finisce con un incolonnamento
di equazioni:
\begin{align}
a &= b + c \\
&= d + e \\
&\qedhere
\end{align}
\end{proof}
```

*Dimostrazione.* Questa è una dimostrazione che finisce con un incolonnamento di equazioni:

$$\begin{aligned} a &= b + c & (3.28) \\ &= d + e & (3.29) \end{aligned} \quad \square$$

Quando le equazioni sono numerate si presenta un problema analogo; si confronti

```
\begin{proof}
Questa dimostrazione termina con
un'equazione.
  \begin{equation}
    a = b + c.
  \end{equation}
\end{proof}
```

con

```
\begin{proof}
Questa dimostrazione termina con
un'equazione.
  \begin{equation}
    a = b + c. \qedhere
  \end{equation}
\end{proof}
```

*Dimostrazione.* Questa dimostrazione termina con un'equazione.

$$a = b + c. \quad (3.30)$$

□

*Dimostrazione.* Questa dimostrazione termina con un'equazione.

$$a = b + c. \quad (3.31)$$

□

e si noti che nella seconda versione (quella corretta) il simbolo □ è molto più vicino all'equazione rispetto all'esempio precedente.

### 3.12 Elenchi di simboli matematici

Le tabelle seguenti mostrano tutti i simboli accessibili con  $\text{\LaTeX}$  prevalentemente in modo matematico. Usando  $\text{\Lua\LaTeX}$  o  $\text{\Xe\LaTeX}$ <sup>8</sup>, qualche simbolo della collezione `amsmath` potrebbe non essere disponibile, rispetto a quanto esposto nelle tabelle seguenti, ma corrispondere a un altro comando; si consulti il documento `unimath-symbols.pdf` che indica tutti i font disponibili con almeno una mezza dozzina di font matematici in formato OpenType.

Nelle tabelle seguenti, i caratteri mancanti non appaiono accanto ai comandi che dovrebbero produrli; ove possibile, sono sostituiti dal segno □. Tuttavia, compilando con  $\text{\Lua\LaTeX}$  o  $\text{\Xe\LaTeX}$ , quando si presenta un messaggio che dice che il tal segno non è contenuto nella base di  $\text{\LaTeX 2}_\epsilon$  si preme il tasto `S` e si prosegue la compilazione.

In questo paragrafo, componendo con  $\text{\Xe\LaTeX}$  e  $\text{\Lua\LaTeX}$  si sono mostrati i font `XITS Math` che forniscono più font di qualsiasi altra collezione, in totale 2427 caratteri. I font `Latin Modern Math` ne contengono “solo” 1585.

Si noti che alcune tabelle mostrano simboli che si possono ottenere avendo caricato il pacchetto `amssymb` nel preambolo del documento.<sup>9</sup>

<sup>8</sup>Si è già accennato a questi due programmi di composizione nel paragrafo 2.5. Oltre a gestire le lingue in modo diverso da quanto fa  $\text{\pdf\LaTeX}$ , essi gestiscono i font matematici in modo più efficiente. Ma per tutti e tre, il linguaggio di mark up è sempre  $\text{\LaTeX}$ .

<sup>9</sup>Le tabelle sono state ottenute da un file `symbols.tex` di David Carlisle, oggi apparentemente non più reperibile, dopo averle modificate profondamente come indicato da Josef Tkadlec. Organizzate diversamente, tabelle del genere si trovano anche in diverse

Tabella 3.2: Accenti matematici

$\hat{a}$	<code>\hat{a}</code>	$\check{a}$	<code>\check{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>
$\grave{a}$	<code>\grave{a}</code>	$\dot{a}$	<code>\dot{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>
$\bar{a}$	<code>\bar{a}</code>	$\vec{a}$	<code>\vec{a}</code>	$\widehat{AAA}$	<code>\widehat{AAA}</code>
$\acute{a}$	<code>\acute{a}</code>	$\breve{a}$	<code>\breve{a}</code>	$\widetilde{AAA}$	<code>\widetilde{AAA}</code>
$\mathring{a}$	<code>\mathring{a}</code>				

Tabella 3.3: Lettere greche

Non ci sono comandi per alcune lettere maiuscole, come per esempio `\Alpha`, `\Beta` e altre, perché sono identiche alle normali maiuscole latine.

$\alpha$	<code>\alpha</code>	$\theta$	<code>\theta</code>	$o$	<code>o</code>	$v$	<code>\upsilon</code>
$\beta$	<code>\beta</code>	$\vartheta$	<code>\vartheta</code>	$\pi$	<code>\pi</code>	$\phi$	<code>\phi</code>
$\gamma$	<code>\gamma</code>	$\iota$	<code>\iota</code>	$\varpi$	<code>\varpi</code>	$\varphi$	<code>\varphi</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\rho$	<code>\rho</code>	$\chi$	<code>\chi</code>
$\epsilon$	<code>\epsilon</code>	$\lambda$	<code>\lambda</code>	$\varrho$	<code>\varrho</code>	$\psi$	<code>\psi</code>
$\varepsilon$	<code>\varepsilon</code>	$\mu$	<code>\mu</code>	$\sigma$	<code>\sigma</code>	$\omega$	<code>\omega</code>
$\zeta$	<code>\zeta</code>	$\nu$	<code>\nu</code>	$\varsigma$	<code>\varsigma</code>		
$\eta$	<code>\eta</code>	$\xi$	<code>\xi</code>	$\tau$	<code>\tau</code>		
$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>	$\Psi$	<code>\Psi</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>	$\Omega$	<code>\Omega</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>		

Tabella 3.4: Relazioni binarie

I seguenti simboli possono essere “negati” facendoli precedere dal comando `\not`.

$<$	<code>&lt;</code>	$>$	<code>&gt;</code>	$=$	<code>=</code>
$\leq$	<code>\leq</code> or <code>\le</code>	$\geq$	<code>\geq</code> or <code>\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> <sup>a</sup>	$\sqsupset$	<code>\sqsupset</code> <sup>a</sup>	$\bowtie$	<code>\bowtie</code> <sup>a</sup>
$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\bowtie$	<code>\bowtie</code>
$\in$	<code>\in</code>	$\ni$	<code>\ni</code> , <code>\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</code> o <code>\ne</code>

<sup>a</sup>Si usi il pacchetto `latexsym` solo nel caso in cui non si sia usato il pacchetto `amssymb`. Alcuni simboli potrebbero mancare se si compila con `XYLaTeX` o `LuaLaTeX`.

Tabella 3.5: Operatori binari

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

<sup>a</sup>Si usi il pacchetto `latexsym` solo nel caso in cui non si sia usato il pacchetto `amssymb`. Alcuni simboli potrebbero mancare se si compila con `XYLaTeX` o `LuaLaTeX`.

Tabella 3.6: Grandi operatori

$\Sigma$	<code>\sum</code>	$\cup$	<code>\bigcup</code>	$\vee$	<code>\bigvee</code>
$\prod$	<code>\prod</code>	$\cap$	<code>\bigcap</code>	$\wedge$	<code>\bigwedge</code>
$\coprod$	<code>\coprod</code>	$\sqcup$	<code>\bigsqcup</code>	$\oplus$	<code>\bigoplus</code>
$\int$	<code>\int</code>	$\oint$	<code>\oint</code>	$\odot$	<code>\bigodot</code>
$\oplus$	<code>\bigoplus</code>	$\otimes$	<code>\bigotimes</code>		

Tabella 3.7: Freccce

$\leftarrow$	<code>\leftarrow</code> or <code>\gets</code>	$\longleftarrow$	<code>\longleftarrow</code>
$\rightarrow$	<code>\rightarrow</code> or <code>\to</code>	$\longrightarrow$	<code>\longrightarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Longleftarrow$	<code>\Longleftarrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</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>
$\lleftarrow$	<code>\lleftarrow</code>	$\rightharpoonup$	<code>\rightharpoonup</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightleftharpoons$	<code>\iff</code> (spazi più grandi)
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\Uparrow$	<code>\Uparrow</code>
$\uparrow$	<code>\uparrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\updownarrow$	<code>\updownarrow</code>	$\nearrow$	<code>\nearrow</code>
$\downarrow$	<code>\downarrow</code>	$\searrow$	<code>\searrow</code>
$\nearrow$	<code>\nearrow</code>	$\nwarrow$	<code>\nwarrow</code>
$\swarrow$	<code>\swarrow</code>		
$\rightsquigarrow$	<code>\leadsto</code> <sup>a</sup>		

<sup>a</sup>Si usi il pacchetto `latexsym` solo nel caso in cui non si sia usato il pacchetto `amssymb`.

Alcuni simboli potrebbero mancare se si compila con `XYLATEX` o `LuaALATEX`.

Tabella 3.8: Grandi frecce e accenti

$\overrightarrow{AB}$	<code>\overrightarrow{AB}</code>	$\underrightarrow{AB}$	<code>\underrightarrow{AB}</code>
$\overleftarrow{AB}$	<code>\overleftarrow{AB}</code>	$\underleftarrow{AB}$	<code>\underleftarrow{AB}</code>
$\overleftrightarrow{AB}$	<code>\overleftrightarrow{AB}</code>	$\underleftrightarrow{AB}$	<code>\underleftrightarrow{AB}</code>

Tabella 3.9: Delimitatori

(	(	)	)	↑	\uparrow
[	[	]	]	↓	\downarrow
{	{	}	}	↕	\updownarrow
⟨	⟨	⟩	⟩	↗	\Uparrow
				↘	\Downarrow
/	/	\	\	↕	\Updownarrow
⌊	⌊	⌋	⌋		
⌈	⌈	⌉	⌉		

Tabella 3.10: Per comporre alcuni grandi delimitatori

{	\lgroup	}	\rgroup	⌋	\lmoustache
	\arrowvert		\Arrowvert	⌋	\bracevert
⌋	\rmoustache				

Tabella 3.11: Altri simboli

...	\dots	...	\cdots	:	\vdots	⋯	\ddots
ℏ	\hbar	ℓ	\imath	ℓ	\jmath	ℓ	\ell
ℜ	\Re	ℑ	\Im	ℵ	\aleph	wp	\wp
∀	\forall	∃	\exists	∅	\emptyset	∂	\partial
'	'	'	\prime	∅	\emptyset	∞	\infty
∇	\nabla	△	\triangle	□	\Box <sup>a</sup>	◇	\Diamond <sup>a</sup>
⊥	\bot	⊤	\top	∠	\angle	√	\surd
◇	\diamondsuit	♥	\heartsuit	♣	\clubsuit	♠	\spadesuit
¬	\neg or \not	♭	\flat	‡	\natural	‡	\sharp

<sup>a</sup>Si usi il pacchetto latexsym solo nel caso in cui non si sia usato il pacchetto amssymb.

Alcuni simboli potrebbero mancare se si compila con X<sub>Y</sub>LaTeX o LuaLaTeX.

Tabella 3.12: Simboli non matematici

Questi simboli possono essere usati anche in modo testo.

†	\dag	§	\S	©	\copyright	®	\textregistered
‡	\ddag	¶	\P	£	\pounds	%	\%

Tabella 3.13:  $\mathcal{AMS}$ : delimitatori accessibili con il pacchetto `amssymb`

$\ulcorner$	<code>\ulcorner</code>	$\urcorner$	<code>\urcorner</code>	$\llcorner$	<code>\llcorner</code>	$\lrcorner$	<code>\lrcorner</code>
$\lvert$	<code>\lvert</code>	$\rvert$	<code>\rvert</code>	$\lVert$	<code>\lVert</code>	$\rVert$	<code>\rVert</code>

Tabella 3.14:  $\mathcal{AMS}$ : lettere greche ed ebraiche accessibili con il pacchetto `amssymb`

$\digamma$	<code>\digamma</code>	$\varkappa$	<code>\varkappa</code>	$\beth$	<code>\beth</code>	$\gimel$	<code>\gimel</code>	$\daleth$	<code>\daleth</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	----------	---------------------	-----------	----------------------

Tabella 3.15: Alfabeti matematici

Alcuni di questi comandi, a pari argomento, producono o meno alcuni segni o ne emettono di sostitutivi a seconda del programma di composizione usato. Per esempio, produrre lettere minuscole con il font fornito da `\mathbb` compilandole con `pdfLATEX`, al posto delle lettere minuscole si ottengono dei segni di relazione binaria, come si vede in questa tabella. `LuaLATEX` e `XLLATEX` non producono questo inconveniente.

Esempio	Comando	Pacchetto richiesto
ABCDEabcde1234	<code>\mathrm{ABCDE abcde 1234}</code>	
<i>ABCDEabcde1234</i>	<code>\mathit{ABCDE abcde 1234}</code>	
$ABCDEabcde1234$	<code>\mathnormal{ABCDE abcde 1234}</code>	
$\mathcal{A}\mathcal{B}\mathcal{C}\mathcal{D}\mathcal{E}$	<code>\mathcal{A}\mathcal{B}\mathcal{C}\mathcal{D}\mathcal{E}</code>	
$\mathscr{A}\mathscr{B}\mathscr{C}\mathscr{D}\mathscr{E}$	<code>\mathscr{A}\mathscr{B}\mathscr{C}\mathscr{D}\mathscr{E}</code>	<code>mathrsfs</code>
$\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{D}\mathfrak{E}\mathfrak{a}\mathfrak{b}\mathfrak{c}\mathfrak{d}\mathfrak{e}1234$	<code>\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{D}\mathfrak{E}\mathfrak{a}\mathfrak{b}\mathfrak{c}\mathfrak{d}\mathfrak{e}1234</code>	<code>amssymb</code>
$\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{D}\mathbb{E}\mathbb{a}\mathbb{b}\mathbb{c}\mathbb{d}\mathbb{e}1234$	<code>\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{D}\mathbb{E}\mathbb{a}\mathbb{b}\mathbb{c}\mathbb{d}\mathbb{e}1234</code>	<code>amssymb</code>

Tabella 3.16:  $\mathcal{AMS}$ : operatori binari accessibili con il pacchetto `amssymb`

$\dot{+}$	<code>\dotplus</code>	$\cdot$	<code>\centerdot</code>		
$\ltimes$	<code>\ltimes</code>	$\rtimes$	<code>\rtimes</code>	$\div$	<code>\divideontimes</code>
$\doublecup$	<code>\doublecup</code>	$\doublecap$	<code>\doublecap</code>	$\smallsetminus$	<code>\smallsetminus</code>
$\veebar$	<code>\veebar</code>	$\bar{\wedge}$	<code>\barwedge</code>	$\overline{\bar{\wedge}}$	<code>\doublebarwedge</code>
$\boxplus$	<code>\boxplus</code>	$\boxminus$	<code>\boxminus</code>	$\ominus$	<code>\circleddash</code>
$\boxtimes$	<code>\boxtimes</code>	$\boxdot$	<code>\boxdot</code>	$\odot$	<code>\circledcirc</code>
$\intercal$	<code>\intercal</code>	$\circledast$	<code>\circledast</code>	$\times$	<code>\rightthreetimes</code>
$\curlyvee$	<code>\curlyvee</code>	$\curlywedge$	<code>\curlywedge</code>	$\times$	<code>\leftthreetimes</code>

Tabella 3.17:  $\mathcal{AMS}$ : Relazioni binarie accessibili con il pacchetto `amssymb`

$\triangleleft$	<code>\lessdot</code>	$\triangleright$	<code>\gtrdot</code>	$\doteqdot$	<code>\doteqdot</code>
$\leqslant$	<code>\leqslant</code>	$\geqslant$	<code>\geqslant</code>	$\risingdotseq$	<code>\risingdotseq</code>
$\leqslantless$	<code>\leqslantless</code>	$\geqslantgtr$	<code>\geqslantgtr</code>	$\fallingdotseq$	<code>\fallingdotseq</code>
$\leqq$	<code>\leqq</code>	$\geqq$	<code>\geqq</code>	$\eqcirc$	<code>\eqcirc</code>
$\lll o \lllless$	<code>\lll o \lllless</code>	$\ggg$	<code>\ggg</code>	$\circ$	<code>\circ</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>	$\approx$	<code>\approx</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>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\Vdash$	<code>\Vdash</code>
$\shortparallel$	<code>\shortparallel</code>	$\Supset$	<code>\Supset</code>	$\Vvdash$	<code>\Vvdash</code>
$\blacktriangleleft$	<code>\blacktriangleleft</code>	$\sqsupset$	<code>\sqsupset</code>	$\backepsilon$	<code>\backepsilon</code>
$\vartriangleright$	<code>\vartriangleright</code>	$\because$	<code>\because</code>	$\varpropto$	<code>\varpropto</code>
$\blacktriangleright$	<code>\blacktriangleright</code>	$\Subset$	<code>\Subset</code>	$\between$	<code>\between</code>
$\trianglerighteq$	<code>\trianglerighteq</code>	$\smallfrown$	<code>\smallfrown</code>	$\pitchfork$	<code>\pitchfork</code>
$\vartriangleleft$	<code>\vartriangleleft</code>	$\shortmid$	<code>\shortmid</code>	$\smallsmile$	<code>\smallsmile</code>
$\trianglelefteq$	<code>\trianglelefteq</code>	$\therefore$	<code>\therefore</code>	$\sqsubset$	<code>\sqsubset</code>

Tabella 3.18:  $\mathcal{AMS}$ : frecce accessibili con il pacchetto `amssymb`

$\dashleftarrow$	<code>\dashleftarrow</code>	$\dashrightarrow$	<code>\dashrightarrow</code>
$\Lleftarrow$	<code>\leftleftarrows</code>	$\Rrightarrow$	<code>\rightrightarrows</code>
$\Leftrightarrow$	<code>\leftrightharpoons</code>	$\Rleftrightharpoons$	<code>\rightleftharpoons</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>
$\multimap$	<code>\multimap</code>	$\Uparrow$	<code>\upuparrows</code>
$\downdownarrows$	<code>\downdownarrows</code>	$\Uparrow$	<code>\upharpoonleft</code>
$\Uparrow$	<code>\upharpoonright</code>	$\Downarrow$	<code>\downharpoonright</code>
$\rightsquigarrow$	<code>\rightsquigarrow</code>	$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>

Tabella 3.19:  $\mathcal{AMS}$ : frecce e relazioni binarie negate accessibili con il pacchetto `amssymb`

$\nless$	$\ngtr$	$\nvarsubsetneqq$
$\lneq$	$\gneq$	$\nvarsupsetneqq$
$\nleq$	$\ngeq$	$\nsubseteqeq$
$\nleqslant$	$\ngeqslant$	$\nsupseteqq$
$\lneqq$	$\gneqq$	$\nmid$
$\lvertneqq$	$\gvertneqq$	$\nparallel$
$\nleqq$	$\ngeqq$	$\nshortmid$
$\lnsim$	$\gnsim$	$\nshortparallel$
$\lnapprox$	$\gnapprox$	$\nsim$
$\nprec$	$\nsucc$	$\ncong$
$\npreceq$	$\nsucceq$	$\nvdash$
$\nprecneqq$	$\nsuccneqq$	$\nvDash$
$\nprecnsim$	$\succnsim$	$\nVDash$
$\nprecnapprox$	$\succnapprox$	$\nVDash$
$\subsetneq$	$\supsetneq$	$\ntriangleleft$
$\varsubsetneq$	$\varsupsetneq$	$\ntriangleright$
$\nsubseteqeq$	$\nsupseteq$	$\ntrianglelefteq$
$\subsetneqq$	$\supsetneqq$	$\ntrianglerighteq$
$\nleftarrow$	$\rightarrow$	$\nleftrightarrow$
$\nLeftarrow$	$\nrightarrow$	$\nLeftrightarrow$

Tabella 3.20:  $\mathcal{AMS}$ : altri simboli accessibili con il pacchetto `amssymb`

$\hbar$	$\hslash$	$\mathbb{k}$
$\square$	$\blacksquare$	$\textcircled{S}$
$\triangleleft$	$\blacktriangle$	$\complement$
$\triangledown$	$\blacktriangledown$	$\Game$
$\lozenge$	$\blacklozenge$	$\bigstar$
$\sphericalangle$	$\measuredangle$	
$\diagup$	$\diagdown$	$\backprime$
$\nexists$	$\Finv$	$\varnothing$
$\eth$	$\sphericalangle$	$\mho^a$

<sup>a</sup>Non deve mai essere usato il simbolo  $\mathcal{U}$  come reciproco dell'unità di misura 'ohm', simbolo  $\Omega$ , ma, in accordo con le norme ISO, si usi solo il 'siemens', simbolo S.

## Capitolo 4

# Funzionalità speciali

Quando si costruisce un lungo documento,  $\LaTeX$  aiuta con alcune funzionalità speciali come la generazione di uno o più indici analitici, di bibliografie e metodi di citazione professionali. Si può avere una descrizione più completa di queste speciali funzionalità e delle estensioni di  $\LaTeX$  in  *$\LaTeX$  Manual* [1] e *The  $\LaTeX$  Companion* [3].

### 4.1 Bibliografia

Si produce la bibliografia con l'ambiente `thebibliography`.

```
\begin{thebibliography}{\langle stringa \rangle}
\langle riferimento1 \rangle
\langle riferimento2 \rangle
...
\end{thebibliography}
```

Ogni  $\langle riferimento \rangle$  comincia con

```
\bibitem[\langle etichetta \rangle]{\langle marker \rangle}
```

Il  $\langle marker \rangle$  è poi usato per citare l'opera, o il libro, o l'articolo, nel documento.

```
\cite{\langle marker \rangle}
```

Se non si usa l' $\langle etichetta \rangle$  opzionale, i riferimenti verranno etichettati con una numerazione progressiva, altrimenti con  $\langle etichetta \rangle$ . Il parametro

`<stringa>` serve per prendere la misura della più lunga etichetta o del numero più lungo che viene usato per marcare (e citare) ogni opera.<sup>1</sup>

```
Knuth~\cite{art:Kn} ha scritto
che \dots
\begin{thebibliography}{9}
\bibitem{art:Kn} E.D.~Knuth
\emph{Titolo}
\TB, vol.~28, num.~3, (2015)
\end{thebibliography}
```

Knuth [1] ha scritto che ...

## Bibliografia

[1] E.D. Knuth *Titolo* TUGboat, vol. 28, num. 3, (2015)

Per progetti più ampi è meglio riferirsi al programma Bib<sub>T</sub>E<sub>X</sub> (nato insieme al L<sub>A</sub>T<sub>E</sub>X), che è incluso in tutte le distribuzioni del sistema T<sub>E</sub>X aggiornate e complete. Esso permette di gestire un database bibliografico estraendone solo le opere citate in un dato documento. L'aspetto grafico di una bibliografia creata con Bib<sub>T</sub>E<sub>X</sub> si basa sul concetto di file di stile; questi file permettono di creare bibliografie conformi a un ampio spettro di forme consolidate. Meglio ancora, si può ricorrere alle recenti aggiunte: il programma `biber` e il pacchetto `biblatex`, che da soli possono produrre una varietà di stili molto più ampia di quelli che si possono ottenere con Bib<sub>T</sub>E<sub>X</sub>.

## 4.2 Indici analitici

Una funzionalità utilissima in molti libri è la presenza di uno o più indici analitici. Con L<sub>A</sub>T<sub>E</sub>X e il programma esterno `makeindex` si può generare un indice analitico molto facilmente. In questa introduzione si descrivono le basi della generazione di tali indici. Per una visione più approfondita si suggerisce di riferirsi al *The L<sub>A</sub>T<sub>E</sub>X Companion* [3]. Più modernamente si può ricorrere al pacchetto `imakeidx` che consente di creare simultaneamente uno o più indici analitici e di comporli in modo sincrono.

<sup>1</sup>Il comando `\TB` può essere definito in uno di pacchetti richiesti nel preambolo, oppure può essere definito dall'utente; serve per scrivere il titolo della rivista TUGboat con un carattere lineare.

Tabella 4.1: Indice analitico: sintassi e alcuni esempi

Esempio	Lemma	Commento
<code>\index{ciao}</code>	ciao, 1	lemma semplice
<code>\index{ciao!Mario}</code>	Mario, 3	lemma secondario sotto ‘ciao’
<code>\index{Lele@\textsl{Lele}}</code>	<i>Lele</i> , 2	lemma formattato
<code>\index{Dani@\textbf{Dani}}</code>	<b>Dani</b> , 7	lemma formattato
<code>\index{Caesar@\textit{Cæsar}}</code>	<i>Cæsar</i> , 33	lemma formattato
<code>\index{bonta@bontà}</code>	bontà, 4	chiave diversa dal lemma
<code>\index{Isa textbf}</code>	Isa, <b>3</b>	numero di pagina formattato
<code>\index{Joe textit}</code>	Joe, 5	numero di pagina formattato

Per attivare questo tipo di indicizzazione di L<sup>A</sup>T<sub>E</sub>X, occorre caricare il pacchetto `makeidx` con

```
\usepackage{makeidx}
```

Esso permette di disporre degli strumenti per raccogliere i dati da inserire nell’indice; bisogna attivarli inserendo nel preambolo il comando

```
\makeindex
```

Il contenuto dell’indice analitico viene specificato mediante comandi del tipo

```
\index{⟨chiave⟩@⟨lemma formattato⟩}
```

dove `⟨lemma formattato⟩` apparirà nell’indice e la `⟨chiave⟩` serve per l’ordinamento alfabetico. Il `⟨lemma formattato⟩` è facoltativo. Se manca (e il carattere `@` non si deve più usare) viene usata la `⟨chiave⟩`. Bisogna inserire i comandi `\index` con i loro argomenti nei punti a cui si vuole che l’indice analitico faccia riferimento. La tabella 4.1 spiega la sintassi e mostra alcuni esempi.

Quando il file sorgente viene elaborato da L<sup>A</sup>T<sub>E</sub>X, ogni comando `\index` scrive l’informazione corretta insieme al numero di pagina in un file speciale. Questo file ha lo stesso nome del file sorgente, ma ha estensione differente (`.idx`). Questo file `.idx` deve poi essere elaborato con il programma

`makeindex`:<sup>2</sup>

`makeindex` nome del file sorgente senza estensione

Il programma `makeindex` elabora il file `.idx` e genera il file `.ind` con i lemmi in ordine alfabetico, senza duplicati e con le sequenze di pagine consecutive raggruppate in un ‘intervallo’.  $\LaTeX$  deve essere eseguito ancora una volta affinché l’indice composto venga stampato dove si trova il comando:

`\printindex`

Si noti che il comando `\index` può modificare la composizione del testo se non è usato correttamente. Ecco un esempio:

Una parola `\index{parola}`. A  
confronto con `parola\index{parola}`.  
Si noti la posizione del punto fermo.

Una parola . A confronto con parola. Si  
noti la posizione del punto fermo.

### 4.3 Testatine e piedini personalizzati

Il pacchetto `fancyhdr`<sup>3</sup> fornisce alcuni semplici comandi per personalizzare le testatine e i piedini del documento. Si guardi alla testatina di questa pagina per vederne un’applicazione.

Ci vuole qualche truccetto per avere il titolo del capitolo e quello del paragrafo corrente nelle testatine.  $\LaTeX$  svolge questo compito con un procedimento in due fasi. Nelle definizioni delle testatine si usano i comandi `\rightmark` e `\leftmark` per la testatina di destra e quella di sinistra, che devono contenere rispettivamente il titolo del paragrafo corrente e quello del capitolo. I valori di queste macro vengono riscritti ogni volta che vengono eseguiti rispettivamente i comandi `\section` e `\chapter`.

Per avere la massima flessibilità, i comandi `\chapter` e `\section` non ridefiniscono direttamente `\leftmark` e `\rightmark`, ma usano `\chaptermark` e `\sectionmark` (o `\subsectionmark` per la classe `article`) per ridefinire le macro da usare nelle testatine.

Perciò, se si vuole cambiare lo stile del nome del capitolo nella testatina, bisogna ridefinire il comando `\chaptermark`.

La figura 4.1 mostra il setup per il pacchetto `fancyhdr` al fine di avere le testatine come appaiono in questo manualetto. In ogni caso, si suggerisce

<sup>2</sup>Se si usa il pacchetto `imakeidx` queste operazioni sono eseguite automaticamente da  $\LaTeX$ . Si consulti la documentazione con `texdoc imakeidx`.

<sup>3</sup>Già presente in ogni installazione completa e aggiornata del sistema  $\TeX$ .

---

```

\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% Con queste impostazioni ci assicuriamo che le testatine
% con il titolo del capitolo e del paragrafo corrente
% siano composti in lettere minuscole. Di default le classi
% standard li compongono in maiuscolo inclinato.
\renewcommand{\chaptermark}[1]{%
    \markboth{#1}{}}
\renewcommand{\sectionmark}[1]{%
    \markright{\thesection\ #1}}
\fancyhf{} % cancella le impostazioni correnti
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % spazio per il filetto
\fancypagestyle{plain}{%
    \fancyhead{} % scarta testatina e piedino per lo stile plain
    \renewcommand{\headrulewidth}{0pt} % scarta il filetto
}

```

---

Figura 4.1: Esempio di impostazione di fancyhdr

di ricorrere alla sua documentazione mediante il solito comando `texdoc fancyhdr`.

## 4.4 Il pacchetto verbatim

Precedentemente si è descritto l'*ambiente verbatim*. In questo paragrafo si espone il funzionamento del *pacchetto verbatim*, che contiene un rifacimento dell'ambiente originale *verbatim* per aggirarne le limitazioni. Questo di per sé non è sorprendente, ma il pacchetto *verbatim* aggiunge alcune nuove funzionalità, e questo è il motivo per il quale se ne parla in questo capitolo.

Per esempio, il comando

`\verbatiminput{⟨nome del file⟩}`

permette di includere il contenuto del file (esterno) *⟨nome del file⟩* come se fosse all'interno di un ambiente *verbatim*.

Per comporre in modo verbatim esistono anche i pacchetti `fancyvrb` e `listings`, alla cui documentazione si rinvia il lettore.

Siccome il pacchetto `verbatim` è parte del gruppo degli ‘strumenti’, dovrebbe essere presente in ogni distribuzione del sistema  $\text{\TeX}$ .

## 4.5 Installare altri pacchetti

Alcune installazioni parziali del sistema  $\text{\TeX}$  sono già dotate di diversi pacchetti aggiuntivi; le distribuzioni complete (e aggiornate) contengono *tutti* i pacchetti disponibili. In ogni caso il sito <https://www.ctan.org/> contiene tutti i pacchetti che nel tempo sono stati creati per il sistema  $\text{\TeX}$ .

I pacchetti come per esempio `geometry`, `hyphenat`, e molti altri, sono tipicamente composti di due file, uno con estensione `.ins`<sup>4</sup> e uno con estensione `.dtx`. Spesso è presente anche un file `README` (o `README.txt`) con una breve descrizione del pacchetto. Per prima cosa bisogna leggere questo file.

Dopo aver copiato i file del pacchetto nella macchina in uso, bisogna elaborarli in modo che (a) la distribuzione (parziale) installata sia informata del nuovo pacchetto; e (b) si disponga della documentazione. Qui si spiega la prima parte.

1. Se il file `.dtx` *non* è autoestraente, compilare con  $\text{\LaTeX}$  il file `.ins` estraendo il file `.sty` e poi compilare il file `.dtx` (generalmente) con `pdf $\text{\LaTeX}$`  per ottenere la documentazione.
2. Spostare il file `.sty` in una cartella dove il sistema  $\text{\TeX}$  installato nella propria macchina possa trovarlo. Generalmente questa è una sottocartella contenuta in `.../⟨localtexmf⟩/tex/latex/` (gli utenti Windows devono cambiare le barre in barre inverse). Il nome effettivo della cartella `⟨localtex⟩` può variare a seconda dell’installazione del sistema  $\text{\TeX}$ ; spesso l’installazione non ne crea una e tocca all’utente crearsela.
3. Aggiornare il database dei nomi dei file. Il comando per eseguire questo aggiornamento dipende dalla distribuzione del sistema  $\text{\TeX}$  installata: per  $\text{\MiKTeX}$  bisogna eseguire `initexmf --update-fndb`, oppure si usi l’interfaccia grafica;  $\text{\TeX}$  Live non richiede aggiornamento se `⟨localtexmf⟩` è la radice dell’albero personale innestato correttamente nella propria `$HOME`; questa dipende dal sistema operativo: `~` per Linux; `~/Library` per Mac; `C:\Users\⟨username⟩\` per sistemi Windows da Win8 in poi.

---

<sup>4</sup>Il file `.ins` servirebbe per estrarre il codice dal file documentato `.dtx`, ma oggi spesso il file `.dtx` è “autoestraente” e il file `.ins` non è più necessario.

4. Controllare se le compilazioni precedenti hanno prodotto un file `.idx`. Se non c'è tale file, la documentazione non dispone di indice analitico e basta passare direttamente al passo 7.
5. Per generare l'indice analitico, si apra un terminale e si dia il comando

```
makeindex -s gind.ist <nome>
```

dove `<nome>` è quello del file principale senza alcuna estensione.
6. Eseguire di nuovo la compilazione del file `.dtx` affinché il file PDF contenga anche l'indice analitico ben composto.
7. Spostare il file PDF in una sottocartella del proprio albero personale lungo il ramo `.../localtexmf/doc/latex/` e si proceda con il database dei nomi dei file come indicato per il file `.sty`. La lettura di questa documentazione dovrebbe essere eseguibile direttamente con il comando `texdoc <nome>`.

Talvolta fra i file prodotti dalla compilazione del file `.dtx` può esserci anche un file con l'estensione `.glo`. Allora fra i passi 6 e 7 si esegua da terminale il comando:

```
makeindex -s gglo.ist -o <nome>.gls <nome>.glo
```

e poi si ricompili ancora una volta il file `.dtx` prima di eseguire il passo 7

## 4.6 L<sup>A</sup>T<sub>E</sub>X e PDF<sup>5</sup>

PDF è un formato di documento portabile (leggibile su qualunque macchina) che accetta l'ipertesto. Similmente a quello che succede in una pagina web, alcune parole sono marcate come collegamenti ipertestuali. Sono dei collegamenti ad altri punti nello stesso documento o persino a documenti esterni. Se si clicca su questi collegamenti ipertestuali, si viene portati direttamente alla destinazione del link. Nel contesto di L<sup>A</sup>T<sub>E</sub>X, questo significa che tutte le istanze di `\ref` e `\pageref` diventano "hyperlink". In più l'indice generale, l'indice analitico e altre simili strutture diventano collezioni di hyperlink.

La maggior parte delle pagine che si trovano in rete sono scritte in HTML (*HyperText Markup Language*), Questo formato ha due svantaggi importanti quando si scrivono documenti scientifici.

1. L'inclusione di formule matematiche in un documento HTML normalmente non è possibile. Anche se esiste uno standard per farlo, la maggior parte dei browser non lo usa o non dispongono dei font necessari per questo scopo.

---

<sup>5</sup>Sezione originariamente composta da Daniel Flipo.

2. Stampare un documento HTML è possibile ma il risultato dipende molto dalla piattaforma e dal browser. I risultati sono generalmente lontanissimi dalla qualità che ci aspettiamo di ottenere nel mondo  $\text{\LaTeX}$ .

Ci sono stati diversi tentativi per creare traduttori dal linguaggio  $\text{\LaTeX}$  ad HTML. Alcuni erano abbastanza buoni nel senso che erano capaci di produrre pagine web leggibili partendo da un file sorgente  $\text{\LaTeX}$ . Ma tutti prendevano delle scorciatoie per arrivare al risultato. Non appena si usano funzionalità appena un po' più complesse di quelle predisposte nel nucleo di  $\text{\LaTeX}$  e/o si usano pacchetti esterni il tutto cade a pezzi. Gli autori che vogliono mantenere la qualità tipografica dei loro documenti anche quando pubblicano sul web, si rivolgono al formato PDF (*Portable Document Format*) che conserva il layout del documento e permette la “navigazione” mediante i collegamenti ipertestuali. La maggior parte dei browser moderni sono già dotati di estensioni per mostrare direttamente i documenti PDF.

Tutti i motori di composizione del sistema  $\text{\TeX}$  possono generare direttamente file in formato PDF. Se si è letta questa guida fino a questo punto, si ha già abbastanza familiarità con questo processo.

#### 4.6.1 Collegamenti ipertestuali o *hyperlink*

Il pacchetto `hyperref` aggiunge due funzionalità molto utili ai documenti PDF.

1. Il formato della carta è impostato secondo l'opzione espressa nella chiamata alla classe.
2. Tutti i riferimenti incrociati del documento diventano hyperlink.

Basta aggiungere `\usepackage{hyperref}` come ultimo comando nel preambolo del file sorgente.

Sono disponibili molte opzioni per personalizzare il comportamento del pacchetto `hyperref`:

- o mediante una lista di opzioni separate da virgole nella chiamata del pacchetto: `\usepackage[opzioni]{hyperref}`; oppure
- dopo aver caricato il pacchetto, con una o più righe di impostazioni della forma `\hypersetup{opzioni}`.

In quel che segue i valori default sono composti in tondo. Inoltre per il programma di lettura si indica uno dei programmi della Adobe, che ha creato lo standard del formato PDF rendendolo pubblico; essa produce i lettori Adobe Reader (gratuito) e Adobe Acrobat (commerciale).

**bookmarks** (`=true,false`) Mostra o nascondi i segnalibri quando si apre il documento.

- unicode** (**=false, true**) Consenti di usare caratteri non appartenenti all'alfabeto latino esteso nei segnalibri del lettore.
- pdftoolbar** (**=true, false**) Mostra o nascondi la barra degli strumenti del lettore.
- pdfmenubar** (**=true, false**) Mostra o nascondi la barra del menu del lettore.
- pdfitwindow** (**=false, true**) Imposta l'ingrandimento iniziale con cui viene aperto il file PDF.
- pdftitle** (**={\<titolo>}**) Definisci il titolo che viene esposto nella finestra Document Info del lettore.
- pdfauthor** (**={\<nome cognome>}**) Imposta il nome dell'autore.
- pdfnewwindow** (**=false, true**) Specifica se il lettore deve aprire un'altra finestra quando un link porta ad un punto fuori del documento corrente.
- colorlinks** (**=false, true**) Circonda i link con una cornice colorata (**false**) o colora direttamente il testo del link (**true**). I colori di questi link possono essere personalizzati mediante le opzioni seguenti (si mostrano i colori di default):<sup>6</sup>
- linkcolor** (**=red**) Colora i link interni (paragrafi, pagine, ecc.).
- citecolor** (**=green**) Colora i link delle citazioni (bibliografia).
- filecolor** (**=magenta**) colora i link ai file.
- urlcolor** (**=cyan**) colora gli URL (email, web).

Se si accettano le impostazioni di default si usi:

```
\usepackage{hyperref}
```

Per avere i segnalibri aperti e i link colorati, ricordando che il valore **true** può essere omissso, si usi:

```
\usepackage[bookmarks,colorlinks]{hyperref}
```

Quando si creano documenti da stampare, i link colorati non sono una buona idea, perché in una stampante in bianco e nero appaiono in grigio, così da essere difficili da leggere. Piuttosto, si usino le cornici colorate che appaiono sullo schermo, ma non vengono stampate.

---

<sup>6</sup>Per questa guida si sono scelti altri colori un po' più scuri di quelli indicati in questa descrizione.

```
\usepackage{hyperref}
\hypersetup{colorlinks=false}
```

oppure li si imposti di colore nero:

```
\usepackage{hyperref}
\hypersetup{colorlinks,%
  citecolor=black,%
  filecolor=black,%
  linkcolor=black,%
  urlcolor=black,%
  pdftex}
```

Se si vogliono aggiungere informazioni alla sezione Document Info del file PDF, si possono usare, per esempio, le seguenti impostazioni.

```
\usepackage[pdauthor={Pierre Desprogres},%
  pdftitle={Des femmes qui tombent},%
  pdftex]{hyperref}
```

In aggiunta ai link automatici per i riferimenti incrociati, è possibile inserire dei link espliciti usando

```
\href{<url>}{<testo>}
```

Il codice

```
Il sito \href{http://www.ctan.org}{CTAN}.
```

produce in uscita “CTAN”; cliccando sulla parola “CTAN”, il lettore visualizza la pagina web indicata.

Se la destinazione di un link non è un URL ma un file locale, si può usare ancora `\href` ma senza specificare la parte `http://`; per esempio:

```
Il documento completo è \href{manuale.pdf}{qui}
```

che produce il testo “Il documento completo è **qui**”. Un clic sulla parola “**qui**” fa sì che il visualizzatore apra il file `manuale.pdf` (il file collegato è relativo al presente documento; se si vuole collegare un file che si trova altrove, bisogna comunque specificare il suo indirizzo relativo alla cartella del presente documento, oppure l’indirizzo assoluto del file).

L’autore di un articolo potrebbe volere che i lettori possano spedirgli dei messaggi di posta elettronica; egli userebbe `\href` dentro l’argomento del comando `\author` nella pagina del titolo del suo documento:

```

\author{Mario Rossi
  <\href{mailto:mario.rossi@server.domain}%
    {mailto:mario.rossi@server.domain}%
  >}

```

Si noti che si è inserito nel comando anche l'indirizzo email che appare sia come link, sia nella pagina stessa dove appare il nome dell'autore. Se si fosse scritto

```
\href{mailto:mario.rossi@server.domain}{Mario Rossi}
```

il link avrebbe funzionato bene nel lettore, ma a stampa l'indirizzo email non sarebbe stato visibile.

### 4.6.2 Problemi con i link

Messaggi come questo:

```

! pdfTeX warning (ext4): destination with the same
  identifier (name{page.1}) has been already used,
  duplicate ignored

```

vengono emessi quando un contatore viene reimpostato, per esempio quando si reimposta il contatore delle pagine a 1 dopo aver composto la pagina del titolo, e si vuole ricominciare a numerare le pagine da 1; in questo caso l'informazione "page 1" non sarebbe più univoca; perciò viene emesso il messaggio che il "duplicate has been ignored."

La contromisura consiste nell'usare l'opzione `plainpages=false` fra quelle di `hyperref`. Sfortunatamente questa impostazione risolve il problema solo con i numeri di pagina. Una soluzione più radicale consiste nell'usare l'opzione `hypertexnames=false`, ma questo renderebbe inefficaci tutti i link di pagina nell'indice analitico.

### 4.6.3 Problemi con i segnalibri

Il testo mostrato nei segnalibri nella schermata del visualizzatore non ha l'aspetto che si desidererebbe che avesse. Questo succede perché i segnalibri sono "solo semplice testo" e quindi solo un numero limitato di caratteri è disponibile per il loro testo rispetto a quanto sarebbe disponibile con L<sup>A</sup>T<sub>E</sub>X. Il pacchetto `hyperref` generalmente si accorge di questo problema ed emette un messaggio:

```

Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:

```

Si aggira il problema fornendo una stringa di testo che sostituisce quella che risulta non valida:

```
\texorpdfstring{\langle testo per TEX \rangle}{\langle testo per segnalibro \rangle}
```

Le espressioni matematiche sono la prima causa per questo genere di problemi:

```
\section{\texorpdfstring{$E=mc^2$}{E = mc ** 2}}
```

che cambia il titolo  $E=mc^2$  in “E = mc \*\* 2” nel segnalibro.

Se si scrive il file sorgente con la codifica Unicode e se si usa l’opzione `unicode` per il pacchetto `hyperref` per usare i caratteri Unicode nei segnalibri, si dispone di una varietà di caratteri molto maggiore tramite `\texorpdfstring`.

## 4.7 Lavorare con X<sub>Ǝ</sub>L<sup>A</sup>T<sub>E</sub>X e LuaL<sup>A</sup>T<sub>E</sub>X<sup>7</sup>

Molto di quanto detto nei paragrafi precedenti è valido anche quando si usa X<sub>Ǝ</sub>L<sup>A</sup>T<sub>E</sub>X o LuaL<sup>A</sup>T<sub>E</sub>X.

### 4.7.1 I font

Oltre ai soliti font basati sulle metriche di T<sub>E</sub>X, X<sub>Ǝ</sub>L<sup>A</sup>T<sub>E</sub>X e LuaL<sup>A</sup>T<sub>E</sub>X usano anche qualsiasi font noto al sistema operativo. Se si dispone, per esempio, del font `Linux Libertine` lo si può usare con il codice seguente:

```
\usepackage{fontspec}
\setmainfont[Ligatures=TeX]{Linux Libertine}
```

inserito nel preambolo. Normalmente questo codice consente di usare anche il corsivo, il neretto e il corsivo neretto usando i soliti comandi `\textit` e `\textbf`. Quando il font è di tipo OpenType si ha accesso anche a molte funzionalità che in passato richiedevano l’uso di font virtuali. La caratteristica più importante di questi font risiede nel loro insieme di caratteri molto esteso che, oltre a quelli latini, può contenere anche quelli per il greco e il cirillico, oltre ad altri caratteri molto più esotici, insieme alle loro specifiche legature.

La maggior parte dei font OpenType dispone di almeno due tipi di cifre: quelle normali tutte della stessa altezza, 0123456789, (in italiano spesso dette *maiuscole*), e quelle chiamate *old style* (in italiano spesso chiamate *minuscole*, 0123456789, che si estendono anche sotto la linea di base. Possono contenere cifre proporzionali, dove lo spazio orizzontale occupato dalla cifra “1” è minore di quello occupato dalla cifra “0”; o cifre monospaziate, dove tutte le cifre occupano lo stesso spazio orizzontale e che, quindi, sono molto

<sup>7</sup>Sezione originariamente composta da Axel Kielhorn.

utili per incolonnare dati numerici nelle tabelle. Si possono impostare con i comandi seguenti:

```
\newfontfamily\LLln[Numbers=Lining]{\meta{font}}
\newfontfamily\LLos[Numbers=OldStyle]{\meta{font}}
\newfontfamily\LLlnm[Numbers=Lining,Numbers=Monospaced]{\meta{font}}
\newfontfamily\LLosm[Numbers=OldStyle,Numbers=Monospaced]{\meta{font}}
```

Quasi tutti i font OpenType contengono le legature standard (ff, fi, fl, ffi, ffl) ma spesso contengono anche rare legature di interesse storico come quelle che legano coppie di lettere come st, ct, tz; Alcune di queste legature sono visibili nella figura 4.2 nella pagina 91. Potrebbero essere fuori luogo in un testo di tipo tecnico-scientifico, ma vanno bene per certi tipi di testi letterari. Per disporre di queste legature si può usare uno dei seguenti comandi.

Pater Noster qui es in caelis: Pater Noster qui es in caelis:  
sanctificétur Nomen Tuum; fanctificétur Nomen Tuum;

Figura 4.2: Legature antiche: a sinistra alcune legature; a destra la ‘s’ lunga e legature ancora più antiche

```
\setmainfont[Ligatures=Rare]{\meta{font}}
\setmainfont[Ligatures=Historic]{\meta{font}}
\setmainfont[Ligatures=Historic,Ligatures=Rare]{\meta{font}}
```

Non tutti i font contengono entrambi i tipi di legature; si consulti la documentazione del font, oppure, più semplicemente, si provi ad usarle e si veda il risultato. Talvolta queste legature dipendono dalla lingua; per esempio, la legatura ‘fk’ è usata in polacco, ma non in inglese. Bisogna, perciò, aggiungere:

```
\setmainfont[Language=Polish]{(font)}
```

Alcuni font (come l’Adobe Garamond Premier Pro, di tipo commerciale) contengono segni alternativi attivati di default da X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X nella versione distribuita con T<sub>E</sub>X Live a partire dal 2010. Il risultato, per esempio, è una lettera “Q” con una coda discendente che sporge a destra fin sotto alla successiva “u”. Per disabilitare questa caratteristica bisogna specificare espressamente la disabilitazione delle caratteristiche *contextuals*.

```
\setmainfont[Contextuals=NoAlternate]{(font)}
```

La maggior parte delle cose dette per X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X vale anche per Lua<sup>A</sup>T<sub>E</sub>X. Per saperne di più, è bene leggere la documentazione del pacchetto fontspec.

### 4.7.2 Dove si trovano i font OpenType?

Con l'installazione di T<sub>E</sub>X Live si dispone di una certa scelta di font OpenType, raccolti nella cartella `.../texmf-dist/fonts/opentype`. Di solito è necessario installarli, cioè di informare il sistema operativo della loro esistenza; questa operazione, ovviamente, dipende dal sistema operativo. Per esempio, la collezione di T<sub>E</sub>X Live non contiene il font DejaVu, che, però, può essere scaricato dal sito <http://dejavu-fonts.org/>. Scaricatolo, bisogna installarlo altrimenti resta inaccessibile.

Si faccia attenzione che ogni font sia installato *una volta sola*, altrimenti possono succedere cose “interessanti”.

Si può usare qualsiasi font presente nel calcolatore, ma bisogna ricordare che altri utenti potrebbero non disporre degli stessi font. Per esempio, il font Zapfino usato negli esempi del pacchetto `fontspec` è disponibile sulle macchine Mac, ma non è disponibile sulle macchine Windows.<sup>8</sup>

### Immettere caratteri Unicode

Il numero di caratteri contenuti in un font è cresciuto, ma il numero di tasti di una normale tastiera è rimasto invariato, perciò, come si immettono i caratteri non accessibili direttamente con la tastiera?

Se si deve scrivere un lungo documento in una lingua straniera, si può installare un driver di tastiera adatto a quella lingua, e stampare le posizioni dei caratteri nei vari tasti. (Molti sistemi operativi hanno una specie di tastiera virtuale; basta prenderne un'immagine dello schermo (*screenshot*).)

Se si deve usare saltuariamente un carattere “esotico”, basta selezionarlo nella tavolozza dei caratteri.

Alcuni ambienti di lavoro (per esempio, il sistema “X Windows”) mettono a disposizione diversi metodi per introdurre caratteri insoliti. Alcuni editor (per esempio, Vim e Emacs) ne mettono a disposizione. Bisogna leggere la documentazione di questi strumenti di lavoro.

### 4.7.3 Compatibilità fra X<sub>Q</sub>L<sup>A</sup>T<sub>E</sub>X o Lua<sup>A</sup>T<sub>E</sub>X e pdf<sup>A</sup>L<sup>A</sup>T<sub>E</sub>X

Ci sono alcune differenze fra pdf<sup>A</sup>L<sup>A</sup>T<sub>E</sub>X e gli altri due programmi di composizione.

- I file sorgente da comporre con X<sub>Q</sub>L<sup>A</sup>T<sub>E</sub>X o Lua<sup>A</sup>T<sub>E</sub>X devono essere salvati con la codifica Unicode (UTF-8) mentre pdf<sup>A</sup>L<sup>A</sup>T<sub>E</sub>X accetta file sorgente salvati con varie codifiche; è buona pratica quella di usare la codifica Unicode anche quando si lavora con pdf<sup>A</sup>L<sup>A</sup>T<sub>E</sub>X.<sup>9</sup>

<sup>8</sup>È però disponibile una versione commerciale chiamata Zapfino Extra.

<sup>9</sup>Dal 2018 è la codifica di default anche per pdf<sup>A</sup>L<sup>A</sup>T<sub>E</sub>X.

- Il pacchetto `microtype` con  $\text{Xe}\text{L}\text{A}\text{T}\text{E}\text{X}$  produce solo la protrusione, mentre con  $\text{pdf}\text{L}\text{A}\text{T}\text{E}\text{X}$  e  $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$  funziona completamente con protrusione e espansione.
- Dal 2018, con i font OpenType la situazione con `microtype` è migliorata molto, ma funziona alla perfezione solo con i font Latin Modern; con altri font le funzionalità sono leggermente inferiori ma ottime lo stesso.

## 4.8 Le presentazioni<sup>10</sup>

Si possono presentare i risultati di un lavoro scientifico proiettando delle “ trasparenze ” su uno schermo mediante un *epidiascopio* (volgarmente detto “lavagna luminosa”); oppure si possono proiettare sullo schermo direttamente dal proprio laptop mediante un apposito proiettore digitale.

I tre programmi  $\text{pdf}\text{L}\text{A}\text{T}\text{E}\text{X}$ ,  $\text{Xe}\text{L}\text{A}\text{T}\text{E}\text{X}$  e  $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$  attraverso la classe `beamer` permettono di creare delle presentazioni in formato PDF in una maniera simile a quella che si potrebbe ottenere con Impress o PowerPoint, se si è in vena di farlo, ma ottenendo un risultato in formato PDF portatile, perché basta portarsi dietro una chiavetta e usare uno dei numerosi sistemi di proiezione che dispongono di un visualizzatore per il formato PDF.

La classe `beamer` usa i pacchetti `graphicx`, `xcolor` e `hyperref` con opzioni adatte alle presentazioni su schermo. Inoltre i font usati sono adatti alle presentazioni.<sup>11</sup>

Quando si compila il codice 4.1 con  $\text{pdf}\text{L}\text{A}\text{T}\text{E}\text{X}$ , si ottiene un file PDF con una prima pagina col titolo e una seconda pagina che mostra un elenco puntato le cui righe vengono mostrate in successione temporale via via che si procede con la presentazione. Con il codice 4.1 si ottengono due quadri divisi in sei schermate, come mostrato nella figura 4.3.

Uno dei vantaggi di `beamer` consiste nel fatto che produce direttamente il file di uscita in formato PDF, usabile direttamente senza bisogno di passare attraverso POSTSCRIPT

Con la classe `beamer` si possono produrre diverse versioni (*modi*) del documento a partire dallo stesso file sorgente. Questo può contenere istruzioni speciali per i diversi modi indicate fra parentesi uncinate. Sono disponibili i seguenti modi.

`beamer` per la presentazione PDF descritta sopra.

`trans` per stampare ogni slide su un foglio di metacrilato (trasparenza).

`handout` per distribuire all’uditorio una versione stampata su carta.

<sup>10</sup>Sezione originariamente composta da Daniel Flipo.

<sup>11</sup>Ancor di più lo sono i font `LXfonts`, pacchetto `LXfonts`, presenti in ogni installazione completa e aggiornata del sistema  $\text{T}\text{E}\text{X}$ ; essi sono derivati da quelli che Knuth ha sviluppato per le sue stesse presentazioni.

```

\documentclass[10pt]{beamer}
\mode<beamer>{%
  \usetheme[hideothersubsections,
            right,width=22mm]{Goettingen}
}

\title{Una presentazione molto semplice}
\author[D. Filippone]{Daniele Filippone}
\institute{U.S.T.L. \& GUT}
\titlegraphic{\includegraphics[width=20mm]{USTL}}
\date{2005}

\begin{document}

\begin{frame}<handout:0>
  \titlepage
\end{frame}

\section{Un esempio}

\begin{frame}
  \frametitle{Cose da fare la domenica pomeriggio}
  \begin{block}{Si potrebbe \ldots}
    \begin{itemize}
      \item passeggiare col cane\ldots \pause
      \item leggere un libro\pause
      \item giocare col gatto\pause
    \end{itemize}
  \end{block}
  e molte altre cose.
\end{frame}
\end{document}

```

Codice 4.1: Esempio di codice per la classe beamer

Il modo di default è `beamer`; lo si cambia semplicemente impostando un modo diverso come opzione globale; per esempio con:

```
\documentclass[10pt,handout]{beamer}
```

nel caso che si vogliono distribuire i fogli stampati.

L'aspetto della presentazione dipende dal tema scelto. Si scelga uno dei temi che accompagnano la collezione di `beamer` o se ne crei uno personalizzato; si veda la documentazione nel file `Beameruserguide.pdf` per maggiori



Figura 4.3: Due quadri con sei schermate

informazioni.

È opportuno dare un'occhiata approfondita al codice [4.1](#).

Usando il modo `<beamer>`, la compilazione della presentazione viene mostrata con il tema `Goettinghen` per disporre del pannello di navigazione integrato con l'indice della presentazione. Le opzioni specificate per questo tema permettono di scegliere le dimensioni del pannello (22 mm in questo esempio) e la sua posizione (a destra del corpo della schermata). L'opzione `hideothersubsections` lascia vedere solo le sottosezioni della sezione corrente. Non ci sono opzioni speciali per la composizione in modo `<transparency>` né per il modo `<handout>`. Queste vengono composte nel

loro layout standard.

I comandi `\title{}`, `\author{}`, `\institute{}` e `\titlegraphic{}` servono per inserire le debite informazioni nel quadro del titolo. Gli argomenti facoltativi dei comandi `\title` e `\author` sono quelli che compaiono nel pannello laterale del tema `Goettingen`.

I titoli e i sottotitoli nel pannello sono creati mediante i normali comandi `\section` e `\subsection`, che però vanno inseriti nel file sorgente *fra e non dentro* i quadri, specificati mediante l'ambiente `frame`.

Le piccole icone al piede di ogni quadro servono anche loro per un'agevole navigazione durante la presentazione; la loro presenza non dipende dal particolare tema scelto.

Il contenuto di ogni quadro viene descritto dentro l'ambiente `frame`. Questo accetta un argomento facoltativo fra parentesi uncinate (`< e >`) per specificare un modo di composizione, così da sopprimere certi quadri in alcuni modi. Nell'esempio, il primo quadro non viene mostrato quando si compone in modo `<handout>` grazie alla specificazione `<handout:0>`.

È fortemente raccomandato di specificare un titolo per ogni quadro successivo al quadro del titolo della presentazione. Questo si può fare sia con un argomento fra parentesi graffe che segue l'apertura dell'ambiente oppure con il comando `\frametitle`; in sostanza, si può scegliere fra queste due sintassi:

```
\begin{frame}{\langle titolo del quadro \rangle}
```

oppure

```
\begin{frame}
\frametitle{\langle titolo del quadro \rangle}
```

Se fosse necessario un sottotitolo, si usi l'ambiente `block` come mostrato nell'esempio. Si noti che i comandi `\section` e `\subsection` non producono nulla dentro i quadri veri e propri.

Il comando `\pause` (nel codice del secondo quadro dell'esempio) permette di mostrare l'elenco puntato una voce alla volta; `beamer` ottiene questo risultato scomponendo il quadro in diverse schermate, come si vede nella figura 4.3, in modo che in ogni successiva schermata dello stesso quadro, quanto è stato esposto nelle schermate precedenti non cambi posizione nelle schermate successive.

In ogni caso, è necessario leggere attentamente la documentazione del pacchetto `beamer` eseguendo da terminale il comando `texdoc beameruserguide` per avere una visione completa di che cosa è disponibile con questo pacchetto. Esso viene continuamente aggiornato e gli aggiornamenti appaiono rapidamente nella distribuzione di `TEX Live`; questo è un motivo in più per mantenere aggiornata l'installazione completa del proprio sistema `TEX`.

## Capitolo 5

# Creare la grafica per la matematica

La maggior parte delle persone usa  $\text{\LaTeX}$  per comporre il testo. Siccome l'approccio di  $\text{\LaTeX}$ , orientato alla struttura, è così conveniente, esso, nella sua versione di base, offre anche delle funzioni, in verità un po' modeste, per produrre risultati grafici da una descrizione testuale. Tuttavia sono state prodotte alcune estensioni per superare la limitatezza delle funzionalità di base. In questo capitolo si esamineranno alcuni di quegli strumenti.

### 5.1 Osservazioni introduttive

$\text{\LaTeX}$  è stato in grado di produrre output grafico fin dall'inizio. Cominciò con l'ambiente `picture` che permette di creare della grafica mettendo in posizione degli elementi predefiniti sulla 'tela' del disegno.  $\text{\LaTeX}$  già conteneva la possibilità di disegnare curve di Bézier quadratiche mediante il comando `\q bezier`, dove `q` richiama la parola "quadratico". Molte curve usate frequentemente come cerchi, ellissi, o catenarie possono venire approssimate in modo soddisfacente dalle curve di Bézier quadratiche, sebbene per farlo ci vogliano un po' di capacità matematiche. Se, però, si usa un linguaggio di programmazione che sia in grado di generare comandi  $\text{\LaTeX}$  contenenti le macro relative alle curve di Bézier, il tutto diventa piuttosto potente.<sup>1</sup>

Nonostante programmare dei disegni in linguaggio  $\text{\LaTeX}$  sia molto limitato, spesso faticoso, tuttavia ci sono ancora dei validi motivi per farlo. I documenti prodotti in questo modo sono "piccoli" in termini di byte, e non c'è nessuna necessità di trascinarsi dietro altri file grafici.

Leslie Lamport stesso, nella seconda edizione del suo manuale, risalente al 1994, aveva indicato un certo numero di macro da usare per evitare le

---

<sup>1</sup>Per esempio, `inkscape` è un programma gratuito con interfaccia grafica che permette di esportare i suoi disegni vettoriali in linguaggio  $\text{\LaTeX}$ -compatibile e con le etichette e le leggende da comporre in  $\text{\LaTeX}$ . Esistono anche altri programmi in grado di farlo.

limitatezze della versione originale di `picture`. Queste estensioni sono state implementate a cominciare dal 2003, e sono proseguite fino ai nostri giorni. Per compatibilità con il passato, non è stato modificato il nucleo di  $\text{\LaTeX}$ , ma è stato messo a disposizione il pacchetto di estensione `pict2e`, che qui daremo sempre per caricato mediante l'istruzione

```
\usepackage{pict2e}
```

inserita nel preambolo del documento. Nella descrizione delle funzionalità svolta nel paragrafo 5.2 si esporranno le funzionalità dell'ambiente `picture` così come sono state estese mediante il pacchetto nominato sopra.

Merita però segnalare subito un grosso vantaggio: il pacchetto `pict2e` produce grafica vettoriale con i compilatori la cui uscita è in formato PDF: `pdf\LaTeX`, `Lua\LaTeX` e `X\LaTeX`; con il solo  $\text{\LaTeX}$  l'uscita è in formato DVI, ma contiene già tutte le istruzioni necessarie per trasformare detto formato nel formato PDF, passando eventualmente per il formato PostScript, entrambi di natura vettoriale.

Questa era la situazione fino ad alcuni anni fa, quando Till Tantau, noto anche per il suo pacchetto<sup>2</sup> `beamer` (per produrre presentazioni da proiettare in aula o in una sala per conferenze) creò il *Portable Graphics Format* (sigla: PGF) con il pacchetto `pgf` e i suoi componenti principali `pgfplots` e `TikZ`, che permettono di creare grafica vettoriale con pieno supporto per diversi programmi di composizione del sistema  $\text{\TeX}$ , compresi i tre programmi basati su  $\text{\LaTeX}$ : `pdf\LaTeX`, `X\LaTeX`, e `Lua\LaTeX`.

Su queste basi sono stati creati molti pacchetti di estensione per scopi precisi. Una grande varietà di queste estensioni sono descritte in dettaglio in *The \LaTeX Graphics Companion* [4].

Forse lo strumento più avanzato collegato con il sistema  $\text{\TeX}$  è METAPOST. È un programma a sé stante basato su METAFONT di Donald E. Knuth. METAPOST usa il linguaggio molto potente e matematicamente sofisticato di METAFONT, ma contrariamente a questo, produce file in formato *encapsulated* POSTSCRIPT che possono essere importate sia con il programma di composizione  $\text{\LaTeX}$ , sia con `pdf\LaTeX`, `X\LaTeX`, e `Lua\LaTeX`. Per una introduzione a METAPOST si veda *A User's Manual for METAPOST* [14], o il testo introduttivo [16].

## 5.2 L'ambiente `picture`<sup>3</sup>

Come ricordato sopra, l'ambiente `picture` fa parte del nucleo di  $\text{\LaTeX}$  è molto adatto per compiti semplici e per controllare con precisione la posizione

<sup>2</sup>Ricordando l'ambiguità della parola 'pacchetto', qui si ricorda che il pacchetto `beamer`, collezione di molti file diversi, contiene la classe `beamer`.

<sup>3</sup>Sezione originariamente composta da Urs Oswald

di ogni elemento sulla pagina. Ma se si deve fare qualunque lavoro serio di disegno, bisogna rivolgersi a `TikZ`, presentato nel paragrafo 5.3 nella pagina 113.

### 5.2.1 Comandi di base

Un ambiente `picture`<sup>4</sup> viene creato con uno dei seguenti due comandi:

```
\begin{picture}(\langle x, y \rangle)...\end{picture}
```

oppure

```
\begin{picture}(\langle x, y \rangle)(\langle x_0, y_0 \rangle)...\end{picture}
```

I numeri (decimali fratti<sup>5</sup>)  $x$ ,  $y$ ,  $x_0$ ,  $y_0$  sono i moltiplicatori dell'unità di misura grafica `\unitlength`, che può venire ridefinita in qualunque momento per scalare un disegno, ma non ne si può cambiare il valore all'interno dell'ambiente stesso; le si può assegnare un valore per esempio con un comando

```
\setlength\unitlength{1.2mm}
```

prima di aprire l'ambiente. Il valore di default di `\unitlength` è `1pt`. La prima coppia di valori,  $(x, y)$ , riserva sulla pagina uno spazio rettangolare per il disegno chiamato *tela*. La seconda coppia di valori,  $(x_0, y_0)$ , è facoltativa e specifica le coordinate del vertice inferiore sinistro della tela rispetto all'origine degli assi cartesiani.

La maggior parte dei comandi ha l'una o l'altra di queste forme:

```
\put(\langle x, y \rangle)\langle oggetto \rangle
```

oppure

```
\multiput(\langle x, y \rangle)(\langle \Delta x, \Delta y \rangle)\langle n \rangle\langle oggetto \rangle
```

dove l'*oggetto* è uno dei tanti elementi grafici o testuali che possono venire collocati sulla tela.

<sup>4</sup>L'ambiente `picture` funziona subito, senza bisogno di caricare nessun pacchetto di estensione. Tuttavia, come anticipato, qui si dà per scontato che si sia caricato nel preambolo il pacchetto `pict2e`, sia perché è utile, sia perché il padre di `LATEX` ne ha descritto il funzionamento prima ancora che ne venisse creato il codice. Evidentemente lo considerava necessario.

<sup>5</sup>Attenzione: all'interno dell'ambiente `picture` il separatore decimale *deve essere sempre il punto, mai la virgola*, che ha altri significati all'interno di questo ambiente.

Le curve di Bézier sono un'eccezione. Quelle quadratiche sono disegnate con il comando:

```
\qbezier(\langle x_1, y_1 \rangle) (\langle x_2, y_2 \rangle) (\langle x_3, y_3 \rangle)
```

e quelle cubiche con il comando:

```
\cbezier(\langle x_1, y_1 \rangle) (\langle x_2, y_2 \rangle) (\langle x_3, y_3 \rangle) (\langle x_4, y_4 \rangle)
```

Ma non è vietato metterne il codice nel secondo argomento di un comando `\put` in modo da regolarne con precisione la posizione senza bisogno di modificare tutte tre le coppie di coordinate per le curve quadratiche e tutte quattro le coppie per le curve cubiche.

### 5.2.2 I segmenti

I segmenti si tracciano con il comando `\line`, di cui si specificano con una coppia di valori i coseni direttori, e con un valore scalare la proiezione della lunghezza sull'asse orizzontale (o, solo per i segmenti perfettamente verticali, sull'asse verticale). Idealmente, il segmento ha un punto iniziale nell'origine degli assi e un punto finale; il comando `\put` colloca l'intero segmento muovendo il suo punto iniziale nella posizione specificata con le coordinate inserite nel suo primo argomento. I coseni direttori sono numeri (decimali in generale fratti) che si consiglia di limitare a valori piccoli in modulo, diciamo non superiori a 10, ma specificare  $(0.5, 0.866)$  oppure  $(500, 866)$  produce lo stesso risultato grafico. Senza usare il pacchetto `pict2e`, i coseni direttori sarebbero limitati ai numeri interi primi fra loro e compresi fra  $-6$  e  $+6$ . Nel primo esempio della pagina 101 sono rappresentati tutti i segmenti del primo quadrante con i coseni direttori interi compresi nell'intervallo previsto precedentemente alla disponibilità di `pict2e`; si vede che i vari segmenti non sono distribuiti angularmente in modo uniforme.

Nel secondo esempio della pagina 101, invece, i segmenti sono creati con i coefficienti di direzione calcolati per gli angoli multipli di  $5^\circ$ . L'esempio mostra anche il ruolo diverso che ha il punto decimale, rispetto alla virgola che separa le due coordinate di ogni coppia

I segmenti sono tracciati con il comando:

```
\put(\langle x, y \rangle) {\line(\langle x_1, y_1 \rangle) {\langle lunghezza \rangle}}
```

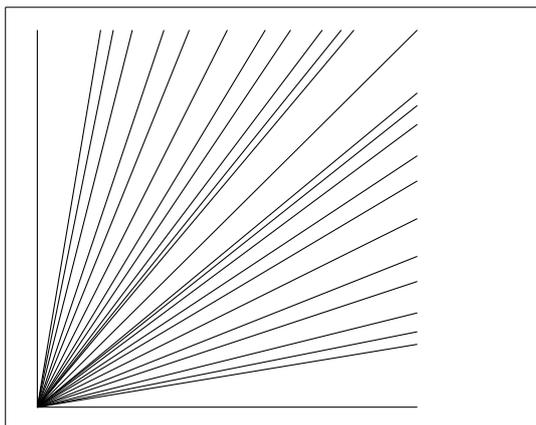
Il comando `\line` vuole due argomenti:

1. un vettore di direzione, e
2. una lunghezza.

```

\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
  \put(0,0){\line(0,1){1}}
  \put(0,0){\line(1,0){1}}
  \put(0,0){\line(1,1){1}}
  \put(0,0){\line(1,2){.5}}
  \put(0,0){\line(1,3){.3333}}
  \put(0,0){\line(1,4){.25}}
  \put(0,0){\line(1,5){.2}}
  \put(0,0){\line(1,6){.1667}}
  \put(0,0){\line(2,1){1}}
  \put(0,0){\line(2,3){.6667}}
  \put(0,0){\line(2,5){.4}}
  \put(0,0){\line(3,1){1}}
  \put(0,0){\line(3,2){1}}
  \put(0,0){\line(3,4){.75}}
  \put(0,0){\line(3,5){.6}}
  \put(0,0){\line(4,1){1}}
  \put(0,0){\line(4,3){1}}
  \put(0,0){\line(4,5){.8}}
  \put(0,0){\line(5,1){1}}
  \put(0,0){\line(5,2){1}}
  \put(0,0){\line(5,3){1}}
  \put(0,0){\line(5,4){1}}
  \put(0,0){\line(5,6){.8333}}
  \put(0,0){\line(6,1){1}}
  \put(0,0){\line(6,5){1}}
\end{picture}

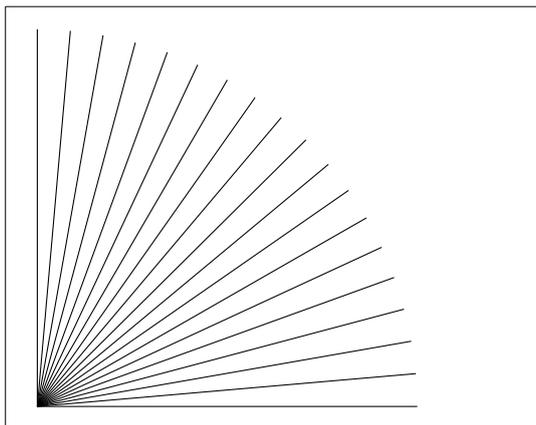
```



```

\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
  \put(0,0){\line(1,0){1}}
  \put(0,0){\line(0.996,0.087){0.996}}
  \put(0,0){\line(0.984,0.173){0.984}}
  \put(0,0){\line(0.965,0.258){0.965}}
  \put(0,0){\line(0.939,0.342){0.939}}
  \put(0,0){\line(0.906,0.422){0.906}}
  \put(0,0){\line(0.866,0.500){0.866}}
  \put(0,0){\line(0.819,0.573){0.819}}
  \put(0,0){\line(0.766,0.642){0.766}}
  \put(0,0){\line(0.707,0.707){0.707}}
  \put(0,0){\line(0.642,0.766){0.642}}
  \put(0,0){\line(0.573,0.819){0.573}}
  \put(0,0){\line(0.500,0.866){0.500}}
  \put(0,0){\line(0.422,0.906){0.422}}
  \put(0,0){\line(0.342,0.939){0.342}}
  \put(0,0){\line(0.258,0.965){0.258}}
  \put(0,0){\line(0.173,0.984){0.173}}
  \put(0,0){\line(0.087,0.996){0.087}}
  \put(0,0){\line(0,1){1}}
\end{picture}

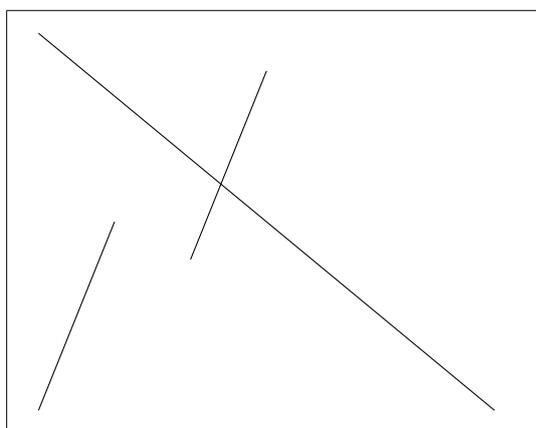
```



In realtà con `pict2e` si dispone di altri comandi per tracciare segmenti e insiemi di segmenti, cioè spezzate e poligoni. Basta specificare le coordinate degli estremi di un solo segmento, o gli estremi dei successivi segmenti; per i poligoni si opera nello stesso modo specificandone solo i vertici e ci pensa il comando a chiudere il poligono congiungendo l'ultimo vertice con il primo della lista.

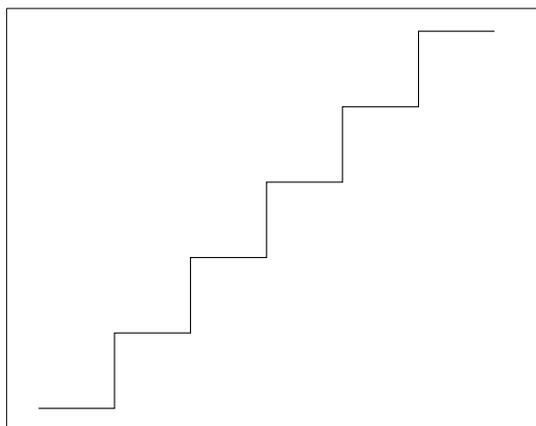
Il comando `\Line` vuole due argomenti: il punto di partenza e il punto di arrivo:

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,5)
\Line(0,0)(1,2.5)
\put(2,2){\Line(0,0)(1,2.5)}
\Line(6,0)(0,5)
\end{picture}
```



Il comando `\polyline` genera una spezzata con un numero arbitrario di segmenti consecutivi; se viene specificato solo un segmento, produce lo stesso effetto di `\Line`.

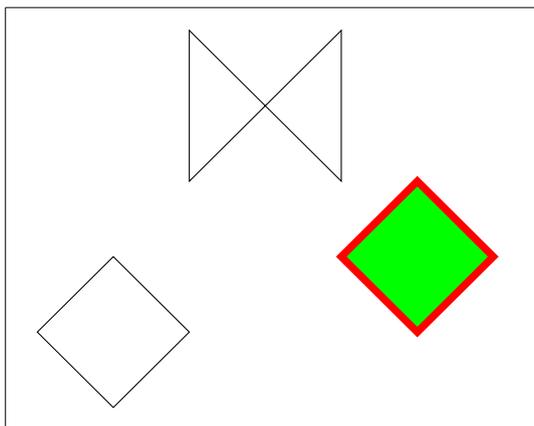
```
\setlength{\unitlength}{1cm}
\begin{picture}(6,5)
\polyline(0,0)(1,0)(1,1)(2,1)%
(2,2)(3,2)(3,3)(4,3)%
(4,4)(5,4)(5,5)(6,5)
\end{picture}
```



Il comando `\polygon` disegna una spezzata chiusa; il comando `\polygon*` definisce il contorno di una spezzata chiusa che riempie con il colore corrente. Il colore può essere specificato con i comandi messi a disposizione dal pacchetto `xcolor` o dal più semplice pacchetto `color`, che devono essere stati caricati (o l'uno o l'altro; preferibilmente il primo) nel preambolo del documento. Se si vuole mostrare anche il contorno di un poligono colorato,

bisogna sovrapporre due poligoni, il primo con il comando asteriscato, il secondo con il comando normale.

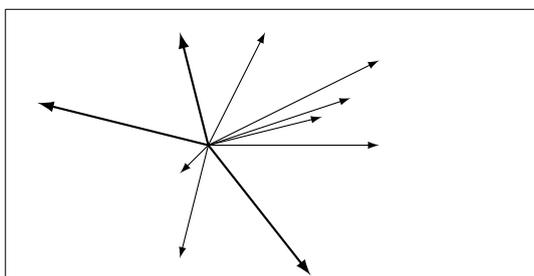
```
\setlength{\unitlength}{1mm}
\begin{picture}(60,50)
% rombo
\polygon(10,0)(20,10)(10,20)(0,10)
% rombo verde con contorno rosso
\put(20,0){\color{green}%
\polygon*(30,10)(40,20)(30,30)(20,20)}
\put(20,0){\linethickness{1mm}%
\color{red}%
\polygon(30,10)(40,20)(30,30)(20,20)}
% fiocco
\polygon(20,30)(40,50)(40,30)(20,50)
\end{picture}
```



### 5.2.3 Frecce

Il comando `\vector` disegna un segmento palesemente orientato sotto forma di una freccia. Senza il pacchetto `pict2e`, i suoi coefficienti di direzione erano limitati ai numeri interi primi fra loro nell'intervallo  $-4 \cdots +4$ . Invece, con questo pacchetto quei coefficienti sono gli stessi coefficienti fratti, che si possono usare con le stesse regole e raccomandazioni date per il comando `\line`; la lunghezza ha lo stesso significato che ha per `\line`. In questo esempio, i coefficienti di direzione sono espressi in entrambi i modi.

```
\setlength{\unitlength}{0.75mm}
\begin{picture}(60,40)
\put(30,20){\vector(1,0){30}}
\put(30,20){\vector(4,1){20}}
\put(30,20){\vector(3,1){25}}
\put(30,20){\vector(2,1){30}}
\put(30,20){\vector(1,2){10}}
\thicklines
\put(30,20){\vector(-4,1){30}}
\put(30,20){\vector(-1,4){5}}
\put(30,20){\vector(0.25,-0.32){18}}
\thinlines
\put(30,20){\vector(-1,-1){5}}
\put(30,20){\vector(-1,-4){5}}
\end{picture}
```



Come si è mostrato nell'esempio, le frecce si disegnano con i comandi

seguenti:

```
\put(\langle x, y \rangle){\vector(\langle x_1, y_1 \rangle){\langle lunghezza \rangle}}
```

Si notino gli effetti dei comandi `\thinlines` e `\thicklines`.

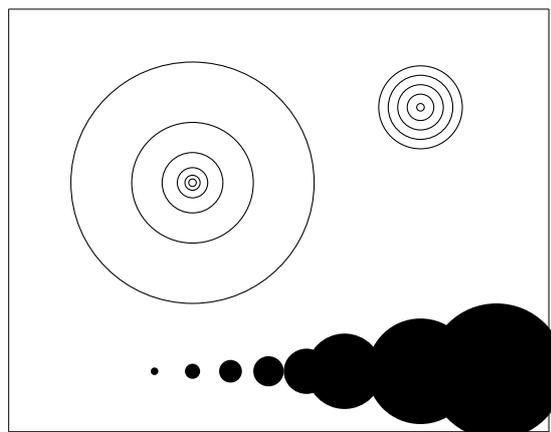
### 5.2.4 Cerchi e circonferenze

I cerchi sono dischi pieni colorati col colore corrente (default: nero) e si disegnano con il comando `\circle*`; le circonferenze sono solo i contorni e si disegnano con il comando `\circle`; per entrambi l'argomento rappresenta il *diametro* del disco o della circonferenza. Con il pacchetto `pict2e`, i diametri possono avere qualunque valore entro limiti ragionevoli, nel senso che il tracciato può uscire dalla tela dichiarata per il disegno, ma non dovrebbe uscire dalla pagina...

```
\setlength{\unitlength}{1mm}
\begin{picture}(60, 50)
  \put(20,30){\circle{1}}
  \put(20,30){\circle{2}}
  \put(20,30){\circle{4}}
  \put(20,30){\circle{8}}
  \put(20,30){\circle{16}}
  \put(20,30){\circle{32}}

  \put(50,40){\circle{1}}
  \put(50,40){\circle{3.5}}
  \put(50,40){\circle{6}}
  \put(50,40){\circle{8.5}}
  \put(50,40){\circle{11}}

  \put(15,5){\circle*{1}}
  \put(20,5){\circle*{2}}
  \put(25,5){\circle*{3}}
  \put(30,5){\circle*{4}}
  \put(35,5){\circle*{6}}
  \put(40,5){\circle*{10}}
  \put(50,5){\circle*{14}}
  \put(60,5){\circle*{18}}
\end{picture}
```



Il comando

```
\put(\langle x, y \rangle){\circle{\langle diametro \rangle}}
```

disegna una circonferenza con il centro in  $(x, y)$  e il  $\langle diametro \rangle$  specificato (non il raggio!). Il comando `\circle*` disegna un cerchio pieno.

Come accennato sopra, il pacchetto `pict2e` permette di disegnare anche curve di Bézier, e archi di vario genere; permette di disegnare curve arbitrarie

con il metodo chiamato in inglese *turtle graphics*: si porta la “penna” in un punto iniziale, poi uno alla volta si aggiungono altri punti connessi con il precedente con segmenti o con curve di Bézier di terzo grado. Infine, arrivati all’ultimo punto si traccia la linea complessiva. Si rinvia il lettore alla documentazione del pacchetto `pict2e`: `texdoc pict2e`.

### 5.2.5 Spessori delle linee

Gli spessori delle linee si fissano con i tre comandi:

```
\thinlines
\thicklines
\linethickness{spessore}
```

Di default `\thinlines` traccia linee spesse `0.4pt`, piuttosto sottili; invece `\thicklines` traccia le linee di spessore doppio, cioè di `0.8pt`; anche se un punto tipografico è circa uguale ad un terzo di millimetro, l’effetto si nota più di quanto si possa immaginare.

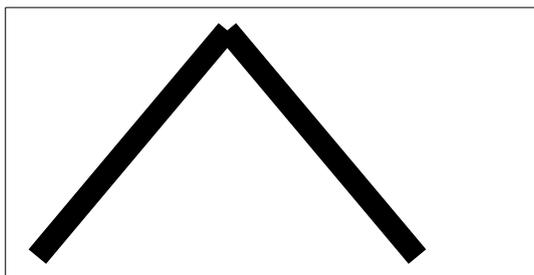
Tuttavia, con `\linethickness` si può specificare uno spessore qualsiasi, esprimendolo in qualunque unità l’utente voglia usare, fra le tante che il sistema `TEX` riconosce; punti (pt), punti didot (dd), pollici (in), millimetri (mm), centimetri (cm), pica (pc), cicero (cc), occhio mediano (ex), corpo (em), persino qualunque altra dimensione conservata in un registro dimensionale, come, per esempio, `\textwidth`.

### 5.2.6 Fine dei percorsi e loro giunzioni

Con linee spesse si vedono chiaramente le giunzioni dei tratti e la terminazione delle linee potrebbe essere troppo brusca; negli esempi che seguono le linee sono spesse 3 mm; le terminazioni e le giunzioni sono esplicitate con i comandi immessi nel codice.

#### Senza specificazioni

```
\setlength{\unitlength}{1mm}
\begin{picture}(60,30)
\linethickness{3mm}
\Line(0,0)(25,30)
\Line(25,30)(50,0)
\end{picture}
```

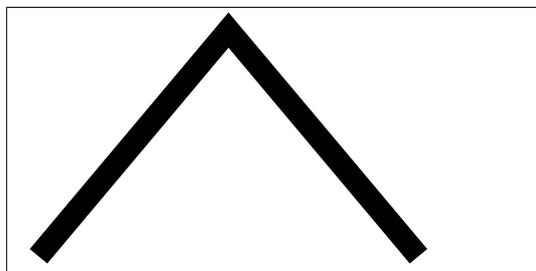


Con `\buttcap` e `\miterjoin`

```

\setlength{\unitlength}{1mm}
\begin{picture}(60,30)
\buttcap\miterjoin
\linethickness{3mm}
\polyline(0,0)(25,30)(50,0)
\end{picture}

```



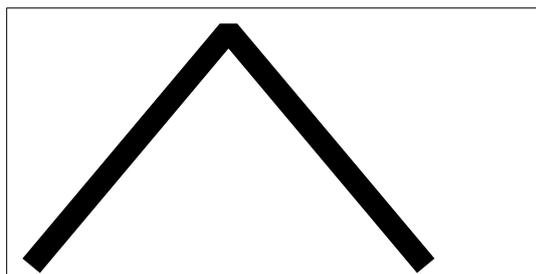
Con linee spesse, si vede chiaramente la convenienza di tracciare una linea continua, anche spezzata, rispetto a tracciare singoli segmenti e archi di curve.

Con `\squarecap` e `\beveljoin`

```

\setlength{\unitlength}{1mm}
\begin{picture}(60,30)
\squarecap\beveljoin
\linethickness{3mm}
\polyline(0,0)(25,30)(50,0)
\end{picture}

```



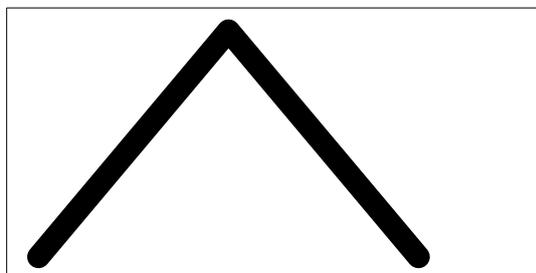
Sfugge ad uno sguardo superficiale, ma si esaminino le terminazioni della linea di questo esempio con quelle dell'esempio precedente: in questo secondo esempio le aste sono prolungate perché la terminazione consiste in uno *square*, più precisamente in un mezzo quadrato.

Con `\roundcap` e `\roundjoin`

```

\setlength{\unitlength}{1mm}
\begin{picture}(60,30)
\roundcap\roundjoin
\linethickness{3mm}
\polyline(0,0)(25,30)(50,0)
\end{picture}

```

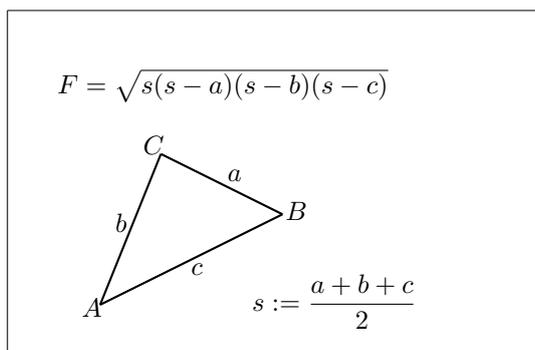


## 5.2.7 Testo e formule

```

\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
  \thicklines
  \put(1,0.5){\line(2,1){3}}
  \put(4,2){\line(-2,1){2}}
  \put(2,3){\line(-2,-5){1}}
  \put(0.7,0.3){\mathbf{A}}
  \put(4.05,1.9){\mathbf{B}}
  \put(1.71,3){\mathbf{C}}
  \put(3.1,2.55){\mathbf{a}}
  \put(1.25,1.7){\mathbf{b}}
  \put(2.5,1){\mathbf{c}}
  \put(0.3,4){\mathbf{F=}}
  \put(3.5,0.4){\mathbf{\displaystyle}}
  \put(3.5,0.4){\mathbf{s:=\frac{a+b+c}{2}}}
\end{picture}

```

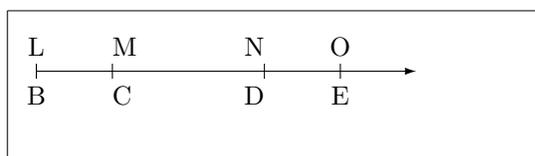


L'esempio mostra che sulla tela si possono mettere non solo dei disegni, ma anche del testo o delle formule. Per collocare questi oggetti non grafici con più precisione, si può ricorrere all'intermezzo di "scatole", come `\makebox` usata in un esempio precedente, o, per incorniciarli, si può usare il comando `\framebox`. La scatola `\makebox` richiede che le siano specificate le dimensioni, anche nulle; possono inoltre essere specificati fino a due codici di posizione scelti fra t (in testa), b (alla base), l (a sinistra), r (a destra) c (al centro). In questo modo, il contenuto viene messo "dentro" la scatola (anche se avesse dimensioni nulle, sporgerebbe fuori, ma è appunto quello che si vuole) in modo da dare una posizione precisa a quel contenuto.

```

\setlength{\unitlength}{1mm}
\begin{picture}(65,10)(0,-5)
  \put(0,0){\vector(1,0){50}}
  \put(0,-1){\line(0,1){2}}
  \put(10,-1){\line(0,1){2}}
  \put(30,-1){\line(0,1){2}}
  \put(40,-1){\line(0,1){2}}
  \put(0,-2){\makebox(0,0)[tc]{B}}
  \put(10,-2){\makebox(0,0)[t1]{C}}
  \put(30,-2){\makebox(0,0)[tr]{D}}
  \put(40,-2){\makebox(0,0)[t]{E}}
  \put(0,2){\makebox(0,0)[bc]{L}}
  \put(10,2){\makebox(0,0)[b1]{M}}
  \put(30,2){\makebox(0,0)[br]{N}}
  \put(40,2){\makebox(0,0)[b]{O}}
\end{picture}

```

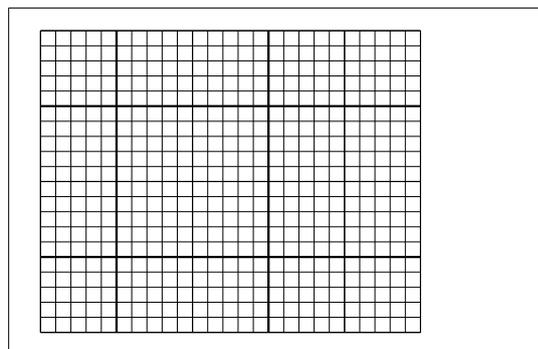


### 5.2.8 `\multiput` e `\linethickness`

```

\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){26}%
    {\line(0,1){20}}
  \multiput(0,0)(0,1){21}%
    {\line(1,0){25}}
  \linethickness{0.15mm}
  \multiput(0,0)(5,0){6}%
    {\line(0,1){20}}
  \multiput(0,0)(0,5){5}%
    {\line(1,0){25}}
  \linethickness{0.3mm}
  \multiput(5,0)(10,0){2}%
    {\line(0,1){20}}
  \multiput(0,5)(0,10){2}%
    {\line(1,0){25}}
\end{picture}

```



Il comando

```
\multiput( $\langle x, y \rangle$ )( $\langle \Delta x, \Delta y \rangle$ ){ $\langle n \rangle$ }{ $\langle oggetto \rangle$ }
```

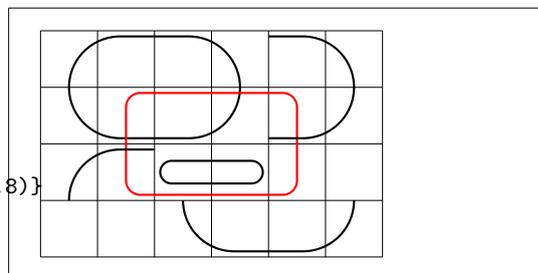
ha quattro argomenti; il punto di partenza, il vettore di traslazione da un oggetto al successivo, il numero *totale* di oggetti, e l'*oggetto* da mettere ripetitivamente in posizione. Si ricordi che `\linethickness`, avendo caricato il pacchetto `pict2e`, agisce su tutte le linee, non solo quelle verticali e orizzontali, come succedeva prima che esso fosse disponibile.

### 5.2.9 Rettangoli arrotondati

```

\setlength{\unitlength}{7.5mm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}%
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}%
    {\line(1,0){6}}
  \thicklines
  \put(2,3){\oval(3,1.8)}
  \put(3,2){\color{red}\oval[0.25](3,1.8)}
  \thicklines
  \put(2,1){\oval(3,1.8)[t1]}
  \put(4,1){\oval(3,1.8)[b]}
  \put(4,3){\oval(3,1.8)[r]}
  \put(3,1.5){\oval(1.8,0.4)}
\end{picture}

```



Il comando

```
\put(\langle x, y \rangle){\oval}[\langle raggio \rangle](\langle w, h \rangle)
```

o

```
\put(\langle x, y \rangle){\oval}[\langle raggio \rangle](\langle w, h \rangle)[\langle parte \rangle]
```

produce un rettangolo con i vertici arrotondati; facoltativamente si può specificare il `\langle raggio \rangle` di curvatura degli spigoli; altrettanto facoltativamente si può specificare quale `\langle parte \rangle` del rettangolo disegnare; si possono specificare fino a due descrittori fra `t` (parte superiore), `b` (parte inferiore), `l` (parte di sinistra), `r` (parte di destra); questa funzionalità era quasi necessaria prima dell'avvento di `pict2e` e consentiva di disegnare alcuni archi di cerchio da 90° o da 180°, ma ora, con `pict2e`, si possono disegnare archi senza limitazioni. Se non si specifica il `\langle raggio \rangle`, viene assunto pari al minore fra metà altezza e metà larghezza del rettangolo; come si vede nell'esempio precedente, dove tutti questi rettangoli sono disegnati con il valore calcolato in automatico, eccetto il rettangolo rosso al centro del disegno.

### 5.2.10 Uso ripetitivo di figure parziali

Talvolta è comodo poter replicare elementi ripetitivi che non siano gestibili con `\multiput`; bisogna salvare certe parti grafiche in “scatole”, e poi usarne il contenuto come se si trattasse di un generico oggetto. L'esempio che segue disegna una “cartella” simile alle icone presenti in molti software; e la usa per costruire un albero di cartelle. Il comando `\savebox` in ambiente `picture` si comporta come `\makebox`, tranne che mette il suo contenuto in una scatola che deve essere già stata precedentemente allocata con un nome:

```
\newsavebox{\langle nome \rangle}
```

poi deve essere riempita con un contenuto:

```
\savebox{\langle nome \rangle}(\langle larghezza, altezza \rangle)[\langle posizione \rangle]{\langle contenuto \rangle}
```

poi il suo `\langle contenuto \rangle` può essere recuperato quando, dove, e quante volte si vuole, con:

```
\put(\langle x, y \rangle){\usebox}{\langle nome \rangle}
```

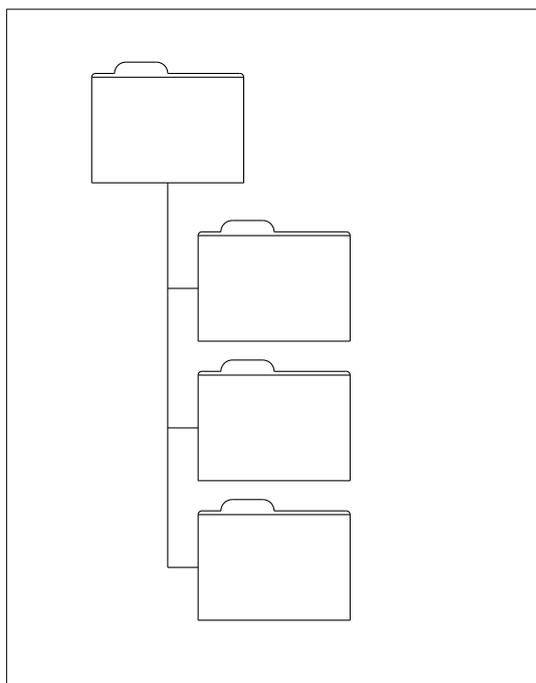
L'argomento facoltativo `\langle posizione \rangle` è formato dai soliti codici che permettono di specificare il punto di riferimento della scatola che il comando `\put` metterà in posizione: `t` (in testa), `b` (alla base), `l` (a sinistra), `r` (a destra); vanno sempre usati a coppie come nell'esempio.

Il  $\langle nome \rangle$  identifica un registro-scatoia di  $\text{\LaTeX}$  e quindi ha la natura di una macro, il che giustifica la barra rovescia nei nomi  $\text{\foldera}$  e  $\text{\folderb}$ . Le figure “in scatolate” possono essere annidate; per esempio la figura della scatola  $\text{\folderb}$  contiene un richiamo della scatola  $\text{\foldera}$ .

```

\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\foldera}
\savebox{\foldera}
  (40,32)[bl]{%      definizione
  \multiput(0,0)(0,28){2}
    {\line(1,0){40}}
  \multiput(0,0)(40,0){2}
    {\line(0,1){28}}
  \put(1,28){\oval(2,2)[tl]}
  \put(1,29){\line(1,0){5}}
  \put(9,29){\oval(6,6)[tl]}
  \put(9,32){\line(1,0){8}}
  \put(17,29){\oval(6,6)[tr]}
  \put(20,29){\line(1,0){19}}
  \put(39,28){\oval(2,2)[tr]}
  }
\newsavebox{\folderb}
\savebox{\folderb}
  (40,32)[l]{% definizione dei folder
    % secondari
    \put(0,14){\line(1,0){8}}
    \put(8,0){\usebox{\foldera}}
  }
\put(34,26){\line(0,1){102}}% albero
  % di cartelle
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
  {\usebox{\folderb}}
\end{picture}

```



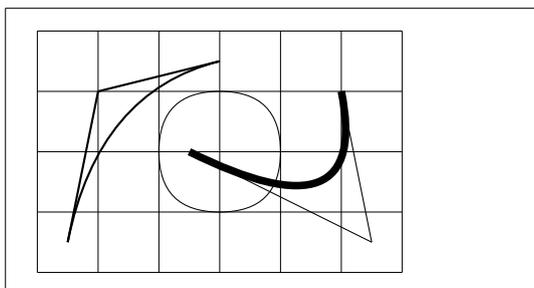
### 5.2.11 Curve di Bézier quadratiche

Qui si mostrano solo alcuni disegni eseguiti con le semplici curve quadratiche.

```

\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}
    {\line(1,0){6}}
  \thicklines
  \put(0.5,0.5){\line(1,5){0.5}}
  \put(1,3){\line(4,1){2}}
  \qbezier(0.5,0.5)(1,3)(3,3.5)
  \thinlines
  \put(2.5,2){\line(2,-1){3}}
  \put(5.5,0.5){\line(-1,5){0.5}}
  \linethickness{1mm}
  \qbezier(2.5,2)(5.5,0.5)(5,3)
  \thinlines
  \qbezier(4,2)(4,3)(3,3)
  \qbezier(3,3)(2,3)(2,2)
  \qbezier(2,2)(2,1)(3,1)
  \qbezier(3,1)(4,1)(4,2)
\end{picture}

```



Come mostra questo esempio, spezzare una circonferenza in quattro archi quadratici di  $90^\circ$  produce risultati insoddisfacenti e bisognerebbe suddividerla almeno in otto archi di  $45^\circ$ ; la stessa operazione con quattro archi di terzo grado, alla base del comando `\circle`, porta invece a un risultato difficilmente distinguibile da una vera circonferenza

### 5.2.12 Catenaria

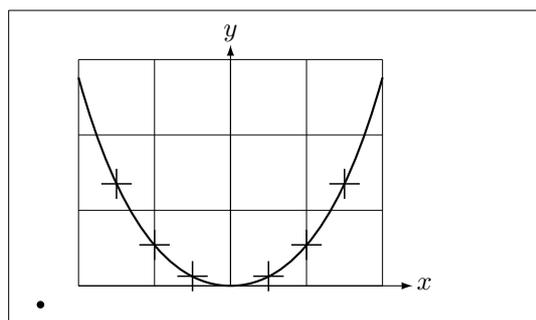
La catenaria è la curva descritta da una catena (da cui il nome) o meglio, da una corda, fermata fra due punti a una certa distanza; risolvendo il problema meccanico, si determina che la curva descritta dalla corda è un arco della curva del coseno iperbolico. Nonostante il coseno iperbolico sia una funzione trascendente, un arco abbastanza grande in prossimità del suo vertice si approssima piuttosto bene con un arco di parabola, quindi con un curva quadratica di Bézier.

L'equazione della catenaria è dunque  $y = \cosh x - 1$ ; è una curva simmetrica rispetto all'asse verticale e ciascuna delle due metà può essere disegnata con una curva di Bézier quadratica; scegliendo opportunamente il punto guida di ciascuno dei due archi si può disegnare un'ottima approssimazione della catenaria; le crocette che compaiono nel disegno sono punti calcolati della catenaria vera, e la differenza fra la curva vera e quella approssimata con la curva di Bézier è impercettibile. Anche questo esempio mostra l'effetto dell'uso del secondo argomento facoltativo del comando di apertura dell'ambiente `picture`. In questo caso l'origine degli assi cartesiani è spostata rispetto al vertice inferiore sinistro della tela: questo vertice ha le coordinate cartesiane  $(-2.5, -0.25)$  e in quel vertice è stato posto il pallino nero.

```

\setlength{\unitlength}{1cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
\put(-2,0){\vector(1,0){4.4}}
\put(2.45,-.05){\mathbb{R}}
\put(0,0){\vector(0,1){3.2}}
\put(0,3.35){\makebox(0,0){\mathbb{R}}}
\put(-2.5,-0.25){\circle*{0.1}}
\thicklines
\qbezier(0.0,0.0)(1.2384,0.0)
(2.0,2.7622)
\qbezier(0.0,0.0)(-1.2384,0.0)
(-2.0,2.7622)
\linethickness{0.075mm}
\multiput(-2,0)(1,0){5}
{\line(0,1){3}}
\multiput(-2,0)(0,1){4}
{\line(1,0){4}}
\linethickness{0.2mm}
\put( 0.3,0.12763){\line(1,0){0.4}}
\put(0.5,-0.07237){\line(0,1){0.4}}
\put(-0.7,0.12763){\line(1,0){0.4}}
\put(-0.5,-0.07237){\line(0,1){0.4}}
\put(0.8,0.54308){\line(1,0){0.4}}
\put(1,0.34308){\line(0,1){0.4}}
\put(-1.2,0.54308){\line(1,0){0.4}}
\put(-1,0.34308){\line(0,1){0.4}}
\put(1.3,1.35241){\line(1,0){0.4}}
\put(1.5,1.15241){\line(0,1){0.4}}
\put(-1.7,1.35241){\line(1,0){0.4}}
\put(-1.5,1.15241){\line(0,1){0.4}}
\end{picture}

```



### 5.2.13 La rapidità nella teoria della relatività speciale

Nella relatività speciale risulta che la velocità riferita a quella della luce segue questa legge

$$\beta = \frac{v}{c} = \tanh \chi$$

Il diagramma si può rappresentare come nell'esempio della pagina 113.

La simmetria della curva permette di eseguire calcoli abbastanza semplici con le curve quadratiche per determinare gli estremi e i punti guida delle curve di Bézier; i calcoli sono decisamente più complessi con le curve cubiche, tuttavia, come si vede dal disegno, la curva rossa (cubica) e la curva nera (quadratica) praticamente si sovrappongono.

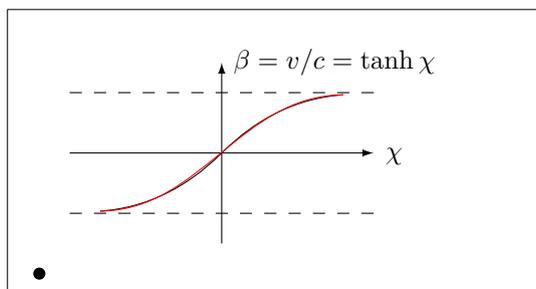
Per disegni più complessi ci si può rivolgere al pacchetto `pstricks` o al pacchetto `tikz`; si veda anche *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion* [4].

Se l'utente se la sente di fare i calcoli necessari e di programmare in linguaggio T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X, l'ambiente `pict2e` permette di fare cose anche più

```

\setlength{\unitlength}{8mm}
\begin{picture}(6,4)(-3,-2)
  \put(-2.5,0){\vector(1,0){5}}
  \put(2.7,-0.1){\mathchi}
  \put(0,-1.5){\vector(0,1){3}}
  \multiput(-2.5,1)(0.4,0){13}
    {\line(1,0){0.2}}
  \multiput(-2.5,-1)(0.4,0){13}
    {\line(1,0){0.2}}
  \put(0.2,1.4)
    {\mathbeta=v/c=\tanh\mathchi}
  \qbezier(0,0)(0.8853,0.8853)
    (2,0.9640)
  \qbezier(0,0)(-0.8853,-0.8853)
    (-2,-0.9640)
    {\color{red}%
  \cbezier(-2,-0.964)(-0.333,-0.950)(0.3,0.950)(2,0.964)}
  \put(-3,-2){\circle*{0.2}}
\end{picture}

```



complicate di quelle esposte qui; benché il testo sia un po' datato, si veda il testo *Graphics in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>* [16].

### 5.3 Le collezioni PGF e TikZ

Oggi ogni compilatore basato su L<sup>A</sup>T<sub>E</sub>X può creare una bella grafica vettoriale; sono solo le interfacce a essere piuttosto diverse rispetto al semplice linguaggio dell'ambiente `picture`. Il pacchetto `pgf` definisce uno strato di astrazione fra le interfacce e l'uscita. Esso viene distribuito con un lungo manuale/tutorial [17]. Perciò, in questo breve paragrafo introduttivo, restiamo molto in superficie.

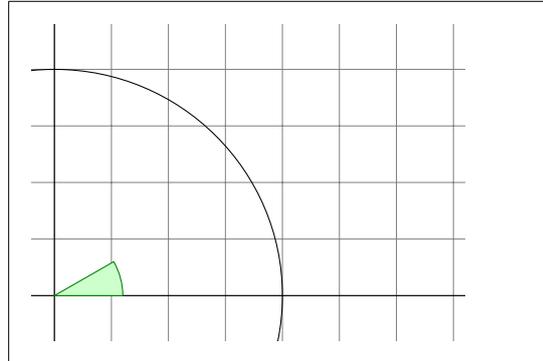
Il pacchetto `pgf` fornisce un linguaggio di alto livello per accedere alle sue funzionalità mediante il pacchetto `tikz`. `TikZ` definisce moltissimi comandi di elevata efficienza per disegnare grafici proprio dall'interno del documento. Bisogna usare l'ambiente `tikzpicture` nel quale si inseriscono i comandi di `TikZ`.

Per prima cosa un esempio con un diagramma privo di senso.

```

\begin{tikzpicture}[scale=3]
  \clip (-0.1,-0.2)
    rectangle (1.8,1.2);
  \draw[step=.25cm,gray,very thin]
    (-1.4,-1.4) grid (3.4,3.4);
  \draw (-1.5,0) -- (2.5,0);
  \draw (0,-1.5) -- (0,1.5);
  \draw (0,0) circle (1cm);
  \filldraw[fill=green!20!white,
    draw=green!50!black]
    (0,0) -- (3mm,0mm)
    arc (0:30:3mm) -- cycle;
\end{tikzpicture}

```

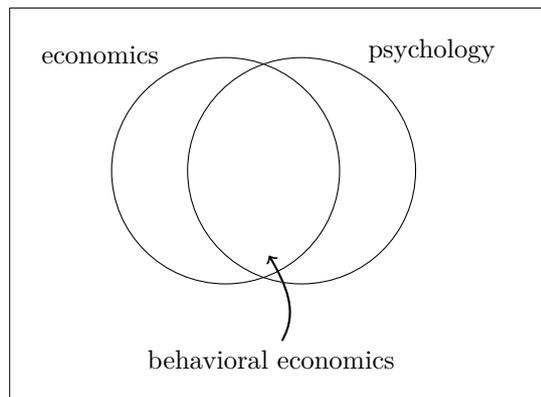


Si noti il punto e virgola usato per terminare il codice dei vari comandi.  
Un semplice diagramma di Venn.

```

\begin{tikzpicture}
  \node[circle,draw,
    minimum size=3cm,
    label=120:{economics}]
    at (0,0) {};
  \node[circle,draw,
    minimum size=3cm,
    label=60:{psychology}]
    at (1,0) {};
  \node (i) at (0.5,-1) {};
  \node at (0.6,-2.5)
    {behavioral economics}
    edge[->,thick,
    out=60,in=-60] (i);
\end{tikzpicture}

```

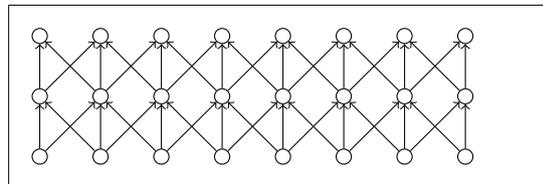


Si notino i cicli foreach nell'esempio seguente.

```

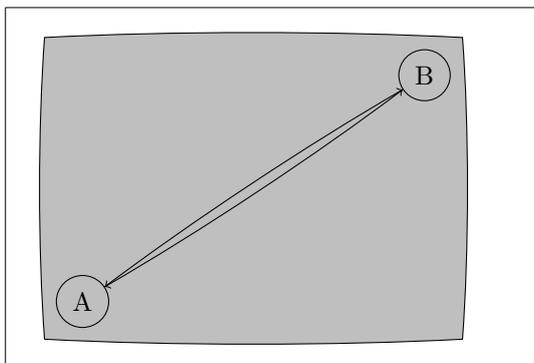
\begin{tikzpicture}[scale=0.8]
  \tikzstyle{v}=[circle, minimum size=2mm,inner sep=0pt,draw]
  \foreach \i in {1,...,8}
    \foreach \j in {1,...,3}
      \node[v]
        (G-\i-\j) at (\i,\j) {};
  \foreach \i in {1,...,8}
    \foreach \j/\o in {1/2,2/3}
      \draw[->]
        (G-\i-\j) -- (G-\i-\o);
  \foreach \i/\n in
    {1/2,2/3,3/4,4/5,5/6,6/7,7/8}
    \foreach \j/\o in {1/2,2/3} {
      \draw[->] (G-\i-\j) -- (G-\n-\o);
      \draw[->] (G-\n-\j) -- (G-\i-\o);
    }
\end{tikzpicture}

```

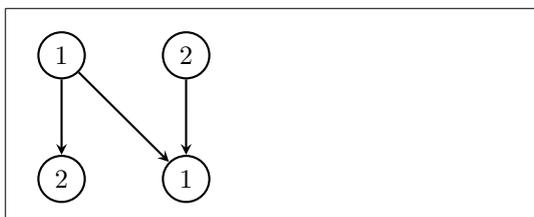


Con l'uso di ulteriori *librerie TikZ* vengono aggiunte un certo numero di funzionalità per disegnare forme speciali, come questa scatola (grigia) con i bordi leggermente curvi.

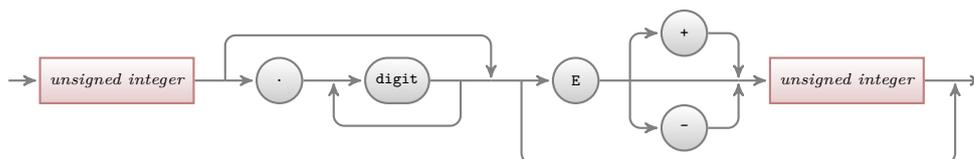
```
\usetikzlibrary{%
  decorations.pathmorphing}
\begin{tikzpicture}[
  decoration={bent,aspect=.3}]
\draw [decorate,fill=lightgray]
(0,0) rectangle (5.5,4);
\node[circle,draw]
(A) at (.5,.5) {A};
\node[circle,draw]
(B) at (5,3.5) {B};
\draw[->,decorate] (A) -- (B);
\draw[->,decorate] (B) -- (A);
\end{tikzpicture}
```



```
\usetikzlibrary{positioning}
\begin{tikzpicture}[xscale=6,
  yscale=8,>=stealth]
\tikzstyle{v}=[circle,
  minimum size=1mm,draw,thick]
\node[v] (a) {$1$};
\node[v] (b) [right=of a] {$2$};
\node[v] (c) [below=of a] {$2$};
\node[v] (d) [below=of b] {$1$};
\draw[thick,->]
(a) to node {} (c);
\draw[thick,->]
(a) to node {} (d);
\draw[thick,->]
(b) to node {} (d);
\end{tikzpicture}
```



Si possono disegnare gli schemi sintattici che hanno l'aspetto di quelli che si trovano in un libro di programmazione Pascal. Il codice intimorisce un poco rispetto a quello degli esempi mostrati sopra, perciò qui si mostra solo il risultato. Se si dà un'occhiata alla documentazione di `pgf` si trova un tutorial nel quale è documentato come disegnare proprio questo diagramma.



E c'è ancora di più; per disegnare dati numerici o andamenti di funzioni è bene esaminare il pacchetto `pgfplots` e la sua documentazione. Può anche

chiamare il programma esterno `gnuplot` per controllare le vere funzioni che si sono disegnate.

Per avere maggiori indicazioni conviene visitare l'eccellente sito di Kjell Magne Fauske <http://www.texample.net/tikz/>. Contiene un deposito continuamente in espansione di grafici bellissimi e altro codice L<sup>A</sup>T<sub>E</sub>X. Su [www.texample.net](http://www.texample.net) si può trovare l'URL del sito [list of examples with PGF/TikZ](#) così che non è necessario scrivere tutto quel codice a mano.

## Capitolo 6

# Personalizzare L<sup>A</sup>T<sub>E</sub>X

I documenti prodotti con i comandi fin qui appresi appariranno sufficienti per un'ampia utenza. Per quanto i documenti così creati non abbiano un aspetto fantasioso, obbediscono a tutte le regole consolidate della buona tipografia, il che li rende facili da leggere e piacevoli da guardare.

Vi sono tuttavia delle situazioni in cui L<sup>A</sup>T<sub>E</sub>X non fornisce un comando o un ambiente che risponde ai nostri intenti, oppure in cui la resa prodotta da certi comandi esistenti non soddisfa le nostre esigenze.

In questo capitolo cercheremo di fornire alcuni suggerimenti su come insegnare a L<sup>A</sup>T<sub>E</sub>X a produrre risultati diversi da quelli predefiniti.

### 6.1 Comandi, ambienti e pacchetti nuovi

Si può notare che tutti i comandi introdotti in questo testo sono stampati all'interno di una cornice, e inoltre che compaiono nell'indice alla fine del libro. Invece di utilizzare direttamente i comandi di L<sup>A</sup>T<sub>E</sub>X necessari a tal fine, è stato creato un pacchetto in cui vengono definiti nuovi comandi e nuovi ambienti per questo scopo. Pertanto si può semplicemente scrivere:

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



\dum

In questo esempio sono stati usati sia un nuovo ambiente chiamato `lscommand`, che è responsabile del disegno della cornice attorno al comando, sia un nuovo comando chiamato `\ci`, che compone il nome del comando e ne inserisce la voce corrispondente nell'indice. Ciò può essere verificato andando a cercare il comando `\dum` nell'indice alla fine di questo libro, dove si troverà una voce per `\dum` che presenta i riferimenti a ogni pagina in cui tale comando è stato menzionato.

Se in un secondo momento si decidesse di non voler più scrivere i comandi racchiusi da una cornice, basterebbe semplicemente modificare la definizione dell'ambiente `lscmmand` per conferirgli un nuovo aspetto. Ciò è molto più efficiente che scorrere l'intero documento per cercare tutti i luoghi in cui sono stati usati dei generici comandi L<sup>A</sup>T<sub>E</sub>X per disegnare le cornici attorno ad alcune parole o frasi.

### 6.1.1 Nuovi comandi

Per aggiungere nuovi comandi personalizzati si usa il comando

```
\newcommand{⟨nome⟩}[⟨num⟩][⟨default⟩]{⟨definizione⟩}
```

Fondamentalmente, il comando richiede due argomenti: il *⟨nome⟩* della macro che si desidera creare e la *⟨definizione⟩* del comando. L'opzione *⟨num⟩* tra parentesi quadre è facoltativa e specifica il numero di argomenti che la nuova macro richiede (fino a un massimo di 9). Se manca, il valore predefinito è 0, cioè nessun argomento. Il secondo argomento facoltativo [*⟨default⟩*] svolge due funzioni: (a) il solo fatto di esistere, anche se *⟨default⟩* fosse vuoto, implica che il primo argomento dei *⟨num⟩* argomenti è facoltativo e (b) che gli si assegna il valore specificato da *⟨default⟩*; dunque, quando si userà il nuovo comando il primo argomento andrà immesso fra parentesi quadre; non ha importanza che cosa sia *⟨default⟩*: è la macro *⟨nome⟩* che saprà cosa farsene.

I due esempi seguenti dovrebbero aiutare a farsi un'idea.

Il primo esempio definisce un nuovo comando chiamato `\tnss`. Esso è l'abbreviazione per “Una (mica tanto) breve introduzione a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>”. Tale comando potrebbe tornare utile se si dovesse scrivere molte volte il titolo di questo libro.

```
\newcommand{\tnss}{Una mica
  tanto breve introduzione a
  \LaTeXe}
Questa è "\tnss" \ldots{\}\
"\tnss"
```

Questa è “Una mica tanto breve introduzione a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>” ...  
 “Una mica tanto breve introduzione a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>”

Il prossimo esempio illustra come definire un nuovo comando che accetta due argomenti, il primo dei quali è facoltativo.

Il parametro `#1` viene sostituito dal primo argomento specificato, `#2` dal secondo e così via.

```
\newcommand{\txsit}[2][mica tanto]
  {Questa è una \emph{#1}
   #2 Introduzione a \LaTeXe}
% nel corpo di testo:
\begin{itemize}
\item \txsit{breve}
\item \txsit[molto]{lunga}
\end{itemize}
```

- Questa è una *mica tanto* breve Introduzione a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>
- Questa è una *molto* lunga Introduzione a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

L<sup>A</sup>T<sub>E</sub>X non consente di creare un nuovo comando che ne sovrascriva uno già esistente. C'è però un comando speciale, nel caso si voglia fare proprio questo: `\renewcommand`. Esso utilizza la stessa sintassi del comando `\newcommand`.

Quando non si sa se il comando esista già o non esista ancora, è conveniente usare il comando `\providecommand`. Esso funziona come `\newcommand` ma, se il comando è già definito, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> tranquillamente non ridefinirà nulla.

Ci sono alcune cose da notare sugli spazi che seguono i comandi di L<sup>A</sup>T<sub>E</sub>X. Per ulteriori informazioni, si veda il paragrafo 6.1.3 nella pagina 120.

### 6.1.2 Nuovi ambienti

Proprio come per il comando `\newcommand`, esiste un comando per creare un ambiente personalizzato. Il comando `\newenvironment` adotta la seguente sintassi:

```
\newenvironment{<nome>}[<num>][<default>]{<prima>}{<dopo>}
```

Come `\newcommand`, `\newenvironment` può ricevere  $\langle num \rangle$  argomenti, dei quali, se è specificato  $[\langle default \rangle]$ , il primo è facoltativo e il suo valore preimpostato vale  $\langle default \rangle$ . Quanto specificato nell'argomento  $\langle prima \rangle$  viene elaborato prima che venga elaborato il codice all'interno dell'ambiente. Quanto specificato nell'argomento  $\langle dopo \rangle$  viene elaborato quando si incontra il comando `\end{<nome>}`.

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

```
\newenvironment{sovrano}%
  {\rule{1ex}{1ex}%
   \hspace{\stretch{1}}%
  {\hspace{\stretch{1}}%
   \rule{1ex}{1ex}}
\begin{sovrano}
Miei umili sudditi\ldots
\end{sovrano}
```

■ Miei umili sudditi... ■

L<sup>A</sup>T<sub>E</sub>X si assicura che non venga definito un ambiente che esiste già. Se però si vuole cambiare un ambiente esistente, si usa il comando `\renewenvironment`. Esso adotta la stessa sintassi del comando `\newenvironment`.

I comandi usati in questo esempio verranno spiegati in seguito. Per il comando `\rule`, vedi pagina 135, per `\stretch` la pagina 127; si possono trovare ulteriori informazioni su `\hspace` alla pagina 127.

### 6.1.3 Spazio extra

Quando si crea un nuovo ambiente, si può facilmente essere infastiditi da spazi extra che si insinuano e che possono potenzialmente produrre effetti fatali, ad esempio quando si intenda creare un ambiente per il titolo che sopprima il proprio rientro nella stessa misura del capoverso che segue. Il comando `\ignorespaces` inserito alla fine del gruppo di comandi *prima* farà ignorare qualsiasi spazio successivo all'esecuzione di quel blocco iniziale. Per il gruppo di comandi *dopo*, la cosa è un po' più complicata, in quanto l'elaborazione speciale avviene alla fine dell'ambiente. Con `\ignorespacesafterend` L<sup>A</sup>T<sub>E</sub>X esegue un `\ignorespaces` dopo che è completata l'elaborazione di `\end{nome}`.

```
\newenvironment{simple}%
  {\noindent}%
  {\par\noindent}

\begin{simple}
Si noti\\lo spazio a sinistra.
\end{simple}
Stessa cosa\\qui.
```

Si noti  
lo spazio a sinistra.

Stessa cosa  
qui.

```
\newenvironment{correct}%
  {\noindent\ignorespaces}%
  {\par\noindent%
   \ignorespacesafterend}

\begin{correct}
Nessuno spazio\\a sinistra.
\end{correct}
Stessa cosa\\qui.
```

Nessuno spazio  
a sinistra.

Stessa cosa  
qui.

### 6.1.4 Pacchetti personalizzati

Quando vengono definiti molti nuovi ambienti e comandi, il preambolo del documento diventerà piuttosto lungo. In questo caso, è una buona idea creare un pacchetto L<sup>A</sup>T<sub>E</sub>X contenente tutte le definizioni dei propri comandi e ambienti. Si usa poi il comando `\usepackage` per rendere il pacchetto disponibile nel documento. Un esempio minimo di codice è contenuto nel codice 6.1 nella pagina 121.

---

```
% Pacchetto demo di Tobias Oetiker, modificato per l'italiano
\ProvidesPackage{demopack}
\newcommand{\tnss}{Una mica tanto breve introduzione
a \LaTeXe}
\newcommand{\txsit}[2]{Una \emph{#1} {#2}
introduzione a \LaTeXe}
\newenvironment{sovrano}{\begin{quote}}{\end{quote}}
```

---

Codice 6.1: Pacchetto di esempio

La scrittura di un pacchetto consiste essenzialmente nel copiare i contenuti del preambolo del documento in un file separato, con estensione `.sty`. C'è un comando speciale

`\ProvidesPackage{(nome pacchetto)}`

da collocare proprio all'inizio del file. `\ProvidesPackage` dice a  $\text{\LaTeX}$  il nome del pacchetto e gli consentirà di emettere un comprensibile avviso quando si tenta di includere un pacchetto due volte. Il codice 6.1 mostra un piccolo pacchetto di esempio che contiene dei comandi simili a quelli definiti negli esempi precedenti.

## 6.2 Caratteri e corpi

### 6.2.1 Comandi per cambiare carattere

$\text{\LaTeX}$  sceglie il carattere appropriato e la dimensione più opportuna in base alla struttura logica del documento (sezioni, note a piè di pagina, ...). In alcuni casi, però, si potrebbero voler cambiare caratteri e dimensioni a mano. Per fare ciò, si utilizzano i comandi elencati nelle tabelle 6.1 e 6.2. La dimensione effettiva di ciascun carattere è un problema di design e dipende dalla classe del documento e dalle sue opzioni. La tabella 6.3 mostra la dimensione assoluta in punti per questi comandi come implementati nelle classi standard dei documenti.

```
{\small I piccoli e
\textbf{coraggiosi} romani
governavano} {\Large per intero
la grande e grossa \textit{Italia}.}
```

I piccoli e **coraggiosi** romani governavano  
 per intero la grande e grossa  
*Italia*.

Una importante caratteristica di  $\text{\LaTeX} 2_{\epsilon}$  è che gli attributi del carattere sono indipendenti. Questo significa che è possibile modificare i corpi e il

carattere stesso conservando gli attributi grassetto o corsivo specificati in precedenza.

Nel *modo matematico* si possono usare i *comandi* atti a cambiare i caratteri per uscire temporaneamente dal *modo matematico* ed entrare nel normale modo testo. Se si vuole modificare il carattere matematico, è necessario ricorrere a un altro speciale insieme di comandi; si veda la tabella 6.4.

Insieme ai comandi relativi alle dimensioni, le parentesi graffe giocano un ruolo importante. Esse vengono usate per costituire *gruppi*. I gruppi servono a limitare il raggio d'azione della maggior parte dei comandi di L<sup>A</sup>T<sub>E</sub>X.

A lui piacciono {\LARGE grandi e  
\small piccole} lettere}.

A lui piacciono grandi e piccole  
lettere.

I comandi relativi alle dimensioni modificano anche la spaziatura fra le righe, ma solo se il capoverso termina all'interno del raggio d'azione del comando stesso. Pertanto, la parentesi deve comparire alla fine del capoverso; se il testo scritto in corpo maggiore del normale è breve e questo testo

Tabella 6.1: Caratteri

<code>\textrm{...}</code>	tondo	<code>\textsf{...}</code>	senza grazie
<code>\texttt{...}</code>	typewriter		
<code>\textmd{...}</code>	medio	<code>\textbf{...}</code>	<b>grassetto</b>
<code>\textup{...}</code>	dritto	<code>\textit{...}</code>	<i>corsivo</i>
<code>\textsl{...}</code>	<i>inclinato</i>	<code>\textsc{...}</code>	MAIUSCOLETTO
<code>\emph{...}</code>	<i>evidenziato</i>	<code>\textnormal{...}</code>	carattere del documento

Tabella 6.2: Dimensioni

<code>\tiny</code>	carattere minuscolo
<code>\scriptsize</code>	carattere piccolissimo
<code>\footnotesize</code>	carattere molto piccolo
<code>\small</code>	carattere piccolo
<code>\normalsize</code>	carattere normale
<code>\large</code>	carattere grande
<code>\Large</code>	carattere più grande
<code>\LARGE</code>	carattere molto grande
<code>\huge</code>	carattere grandissimo
<code>\Huge</code>	carattere enorme

Tabella 6.3: Misure assolute (arrotondate) in punti dei corpi delle classi standard; il corpo normale è definito nell'opzione di classe; quella predefinita è 10 pt.

Corpo	10 pt	11 pt	12 pt
<code>\tiny</code>	5	6	6
<code>\scriptsize</code>	7	8	8
<code>\footnotesize</code>	8	9	10
<code>\small</code>	9	10	11
<code>\normalsize</code>	10	11	12
<code>\large</code>	12	12	14
<code>\Large</code>	14	14	17
<code>\LARGE</code>	17	17	20
<code>\huge</code>	20	20	25
<code>\Huge</code>	25	25	25

Tabella 6.4: Caratteri matematici

<code>\mathrm{...}</code>	Roman Font
<code>\mathbf{...}</code>	<b>Boldface Font</b>
<code>\mathsf{...}</code>	Sans Serif Font
<code>\mathtt{...}</code>	Typewriter Font
<code>\mathit{...}</code>	<i>Italic Font</i>
<code>\mathcal{...}</code>	<i>CALLIGRAPHIC FONT</i>
<code>\mathnormal{...}</code>	<i>Normal Math Font</i>

speciale si trova in linea con il resto del testo, non è necessario usare la graffa chiusa dopo la fine del capoverso. Da notare la posizione del comando `\par`<sup>1</sup> nei prossimi due esempi dove dovrebbe essere chiarissimo che determina esplicitamente la fine del capoverso, se lo si mette dentro o fuori il gruppo; nel secondo caso la spaziatura fra le righe è troppo piccola perché è quella del corpo normale, non quella del corpo `\Large`.

```
{\Large Non leggere questo!  
Non è vero.  
Puoi credermi!}\par}
```

Non leggere questo! Non è vero. Puoi credermi!

```
{\Large Neanche questo è vero.  
Ma ricordati che sono un  
bugiardo.}\par}
```

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

Se si vuole cambiare il corpo di un intero capoverso di testo o anche più, si ricorre alla versione ambiente dei comandi per cambiare carattere.

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

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

Questo risparmia dal dover contare molte parentesi graffe.

### 6.2.2 Pericolo, pericolo, ...

Come rilevato all'inizio di questo capitolo, è pericoloso riempire il documento con comandi espliciti come quelli menzionati finora, perché essi operano in modo antitetico alla filosofia di L<sup>A</sup>T<sub>E</sub>X, che è separare il codice logico da quello estetico del documento. Questo significa che se si usa lo stesso comando per cambiare carattere in luoghi diversi al fine di stampare un tipo speciale di informazione, bisognerebbe usare `\newcommand` per definire una sorta di “involucro logico” per comporre quell'informazione.

```
\newcommand{\oops}[1]{\textbf{#1}}  
Non \oops{entrare} in questa stanza,  
è piena di \oops{macchine}  
con origine e finalità sconosciute.
```

Non **entrare** in questa stanza, è piena di **macchine** con origine e finalità sconosciute.

<sup>1</sup>`\par` equivale a una riga vuota.

Questo approccio ha il vantaggio che si può decidere in una fase successiva di voler usare una rappresentazione estetica di pericolo diversa da `\textbf`; così facendo non è necessario scorrere l'intero documento per trovare tutte le istanze di `\textbf` e quindi verificare se sono state usate per indicare pericolo oppure per qualche altro scopo.

È diverso dire a L<sup>A</sup>T<sub>E</sub>X di *evidenziare* con `\emph` qualcosa e dirgli di usare un *carattere* differente. Il comando `\emph` è sensibile al contesto, laddove i comandi per i caratteri sono assoluti.

```
\textit{È possibile
\emph{evidenziare} un testo
anche se è in corsivo},
\textsf{in un carattere
\emph{senza grazie,}}
\texttt{oppure in stile
\emph{macchina da scrivere}.}
```

*È possibile evidenziare un testo anche se è in corsivo, in un carattere senza grazie, oppure in stile macchina da scrivere.*

### 6.2.3 Consiglio

Per concludere questo viaggio nella terra dei caratteri e dei loro corpi, aggiungiamo qui un breve consiglio:

**Ricordati!** *Quanti più* caratteri **Scegli** di usare in un documento,  
*tanto più LEGGIBILE e bello quel documento diventa.*

## 6.3 Spaziature

### 6.3.1 Scartamento o avanzamento di riga

Se si vuole aumentare lo scartamento<sup>2</sup> di un documento, si modifica il suo valore inserendo nel preambolo il comando

```
\linespread{<fattore>}
```

Si usa `\linespread{1.3}` per ottenere uno scartamento che visivamente sembra “una volta e mezzo” lo scartamento normale, mentre `\linespread{1.6}` per ottenere uno scartamento che appare “doppio” del normale. Normalmente, lo *scartamento* è 1,2 volte l'ammontare del *corpo* del font corrente, e questo spiega perché, per esempio, il fattore 1,3 mostra uno scartamento del 50% superiore al corpo; per riportare lo scartamento al suo valore normale si specifica il *<fattore>* 1. È ancora meglio se si esegue qualunque modifica del *<fattore>* dentro un gruppo che ne limiti il raggio d'azione.

<sup>2</sup>In inglese si chiama *baseline skip* che tradotto letteralmente indica l'avanzamento della riga di base fra due linee di testo consecutive.

L'effetto del comando `\linespread` è piuttosto drastico e non appropriato per un normale testo stampato. Di conseguenza, solo se c'è una buona ragione per cambiare lo scartamento, si usa il comando:

```
\linespread{1.5}\selectfont
```

```
{\linespread{1.5}\selectfont
Questo paragrafo è stampato
con l'avanzamento di riga impostata
a 1.5 rispetto a quello normale.
Da notare il comando \cs{par} alla
fine del capoverso.\par}
```

Questo capoverso ha un chiaro proposito: mostrare come, dopo che la parentesi graffa è stata chiusa, ogni cosa torni normale.

Questo paragrafo è stampato con l'avanzamento di riga impostata a 1.5 rispetto a quello normale. Da notare il comando `\par` alla fine del capoverso.

Questo capoverso ha un chiaro proposito: mostrare come, dopo che la parentesi graffa è stata chiusa, ogni cosa torni normale.

### 6.3.2 Formattazione dei capoversi

In L<sup>A</sup>T<sub>E</sub>X ci sono due parametri che influenzano l'aspetto dei capoversi. Inserendo nel preambolo del file di input una definizione come

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

perché il secondo aumenta lo spazio fra due capoversi, mentre il primo ne annulla il rientro.

I due parametri `plus` e `minus` della lunghezza sopra impostata suggeriscono a T<sub>E</sub>X che può comprimere ed espandere lo spazio fra i paragrafi al massimo della quantità specificata, se serve che i capoversi stessi riempiano completamente la pagina.

Se i capoversi sono separati da un po' di spazio anziché rientrati si segue una pratica del tutto sconsigliabile e che ha effetti collaterali. Infatti, questo produce i suoi effetti anche nell'indice, le cui righe risultano di conseguenza maggiormente spaziate. Per evitare tale effetto, si possono spostare i due comandi dal preambolo all'interno del documento, in un qualche posto al di sotto del comando `\tableofcontents` oppure non usarli affatto, perché la maggior parte dei libri di editoria professionale usa il rientro e non la spaziatura per distinguere i capoversi.

Se si vuole far rientrare un capoverso che normalmente non sarebbe rientrato, si usa

```
\indent
```

all'inizio del capoverso<sup>3</sup>. Ovviamente, questo avrà effetto solo se `\parindent` non è impostato a zero.

Per creare un capoverso privo di rientro, si usa

```
\noindent
```

come primo comando del capoverso. Ciò potrebbe essere utile quando si inizia un documento con del testo, anziché con un comando di sezionamento.

### 6.3.3 Spazio orizzontale

$\LaTeX$  determina automaticamente gli spazi fra parole e frasi. Per aggiungere spazio orizzontale si usa:

```
\hspace{\langle lunghezza \rangle}
```

Se tale spazio dev'essere conservato anche se cade alla fine o all'inizio di una riga, si usa la versione asteriscata `\hspace*`. La  $\langle lunghezza \rangle$ , nel caso più semplice, non è che un numero seguito da un'unità di misura.

Le più importanti unità di misura sono elencate nella tabella 6.5.

Questo `\hspace{15mm}` è uno spazio di 15\,mm.

```
Questo      è uno spazio di 15 mm.
```

Il comando

```
\stretch{\langle fattore \rangle}
```

genera uno speciale spazio elastico. Esso si espande fino a riempire tutto lo spazio residuo di una riga. Se più comandi `\hspace{\stretch{\langle fattore \rangle}}` vengono usati sulla stessa riga, essi occupano tutto lo spazio disponibile in proporzione ai loro rispettivi fattori di espansione.

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

```
x      x      x
```

<sup>3</sup>Per attivare il rientro del primo capoverso dopo i titoli di sezione, si usa il pacchetto `indentfirst`, contenuto nella raccolta 'tools'.



```
\vspace{\stretch{1}}
Questo va all'ultima riga della pagina.\newpage
```

Lo spazio aggiuntivo fra due righe dello *stesso* capoverso o all'interno di una tabella è specificato mediante il comando:

```
\[\langle lunghezza \rangle]
```

Mediante `\bigskip`, `\medskip` e `\smallskip` si aggiunge una quantità predefinita di spazio verticale senza doversi preoccupare di calcolarlo con esattezza.

## 6.4 Impostazione della pagina

$\text{\LaTeX} 2_{\epsilon}$  consente di specificare la dimensione del foglio come opzione al comando `\documentclass`. Esso imposta i margini corretti, ma talvolta i valori predefiniti possono non essere soddisfacenti per un dato documento. Naturalmente, essi possono essere modificati. La figura 6.1 mostra tutti i parametri che possono essere modificati. La figura è stata prodotta grazie al pacchetto `layout`, della raccolta `tools`.<sup>4</sup>

**Un attimo!** Prima di lanciarsi nella frenesia del “Rendiamo un po’ più larga questa pagina così stretta”, è opportuno soffermarsi a riflettere qualche secondo. Come per la maggior parte delle cose in  $\text{\LaTeX}$ , c’è una buona ragione perché l’aspetto della pagina sia così com’è.

Certo, rispetto alla pagina predefinita dei word processor, essa sembra terribilmente stretta. Ma se si dà un’occhiata al proprio libro preferito<sup>5</sup> e si conta il numero di caratteri presenti su una riga di testo standard, si scoprirà che non c’è più di una settantina di caratteri all’incirca per ogni riga. Se si fa lo stesso con una pagina di  $\text{\LaTeX}$ , allo stesso modo si scoprirà che anche qui c’è circa lo stesso numero di caratteri. L’esperienza mostra che la lettura diventa tanto più difficoltosa quanto più numerosi sono i caratteri. È difficile per gli occhi spostarsi dalla fine di una lunga riga fino all’inizio di quella successiva; questo è anche il motivo per cui i giornali sono composti in più colonne relativamente strette.

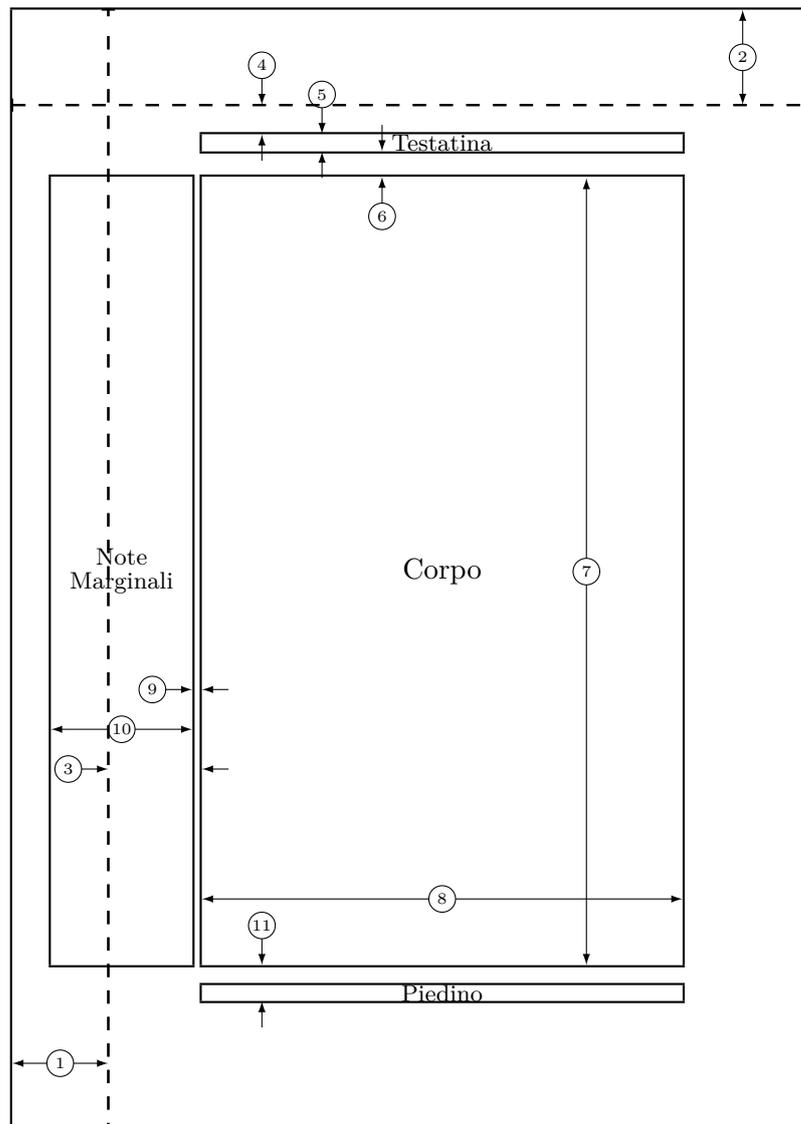
Quindi, se si aumenta la larghezza del corpo di testo, si rende difficile la vita ai lettori.

Ma per non eccedere in cautela, e visto che abbiamo promesso di dire come fa. . .

$\text{\LaTeX}$  fornisce due comandi per modificare questi parametri, che solitamente sono usati nel preambolo del documento.

<sup>4</sup>[CTAN://macros/latex/required/tools](http://ctan://macros/latex/required/tools).

<sup>5</sup>Un vero libro stampato, prodotto da un buon editore.



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

Figura 6.1: Parametri per l'impostazione delle pagine di questa guida. Il pacchetto `layouts` stampa le impostazioni della pagina del documento corrente.

Il primo comando attribuisce un valore fisso a un qualsiasi parametro

dimensionale:

```
\setlength{<parametro>}{<lunghezza>}
```

Il secondo comando aggiunge una determinata lunghezza a un qualsiasi parametro dimensionale:

```
\addtolength{<parametro>}{<lunghezza>}
```

Questo secondo comando è in genere più utile di `\setlength`, perché opera sui valori già impostati. Per ampliare di un centimetro l'ampiezza complessiva del testo si inseriscono nel preambolo del documento i seguenti comandi:

```
\addtolength{\hoffset}{-5mm}
\addtolength{\textwidth}{10mm}
```

In questo ambito è possibile prendere in considerazione il pacchetto `calc`. Esso consente di effettuare operazioni aritmetiche nell'argomento di `\setlength` e in altri luoghi dove, negli argomenti di una funzione, sono inseriti dei valori numerici. Almeno dal 2009, l'estensione  $\varepsilon$ -`TeX` all'interprete `TeX` è stata incorporata in tutti gli altri programmi di composizione, quindi basta consultarne la documentazione per conoscere i nuovi comandi; nel 2018 si è reso disponibile anche il pacchetto `xfp`, che consente di eseguire direttamente i calcoli con numeri reali in virgola mobile (*floating point*).

## 6.5 Altro divertimento con le lunghezze

Dove possibile, è preferibile evitare di usare lunghezze assolute nei documenti `LaTeX`. È meglio basarsi sulla larghezza o sull'altezza di qualche elemento della pagina: ad esempio, la larghezza di una figura potrebbe essere impostata a `\textwidth` per farne riempire orizzontalmente la griglia di stampa.

I tre comandi che seguono consentono di determinare l'ampiezza, l'altezza e la profondità di una stringa di `<testo>`.

```
\settoheight{<sequenza di controllo>}{<testo>}
\settodepth{<sequenza di controllo>}{<testo>}
\settowidth{<sequenza di controllo>}{<testo>}
```

Con `<sequenza di controllo>` si intende una stringa che ha le stesse caratteristiche del nome di un comando, ma che designa, per esempio, un registro-lunghezza; per esempio, la sequenza di controllo `\textwidth` individua il contenuto di un tale registro nel quale è conservato il valore della larghezza

della gabbia del testo; il valore conservato in tali registri si può scrivere nel documento facendone precedere il nome dal comando `\the`.

In questo testo la larghezza della gabbia vale `\texttt{\the\textwidth}`.

In questo testo la larghezza della gabbia vale 181.39386pt.

Nell'esempio appena mostrato si legge che la giustezza vale 181,39386pt; si tratta della giustezza del riquadro dell'esempio; in realtà, usando il comando `\the\textwidth` fuori dell'esempio, il risultato è 360.0pt.

L'esempio che segue mostra una possibile applicazione di questi comandi:

```
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }%
  \makebox[0pt][r]{#1:\ }%
  \ignorespaces}{}%
La formula di Pitagora lega le
misure dei cateti a quella
dell'ipotenusa di un triangolo
rettangolo:
\[
a^2+b^2=c^2
\]

\begin{vardesc}{dove}
$a$, $b$ -- sono i cateti
del triangolo.

$c$ -- è l'ipotenusa del triangolo.

$d$ -- non c'entra niente.
\end{vardesc}
```

La formula di Pitagora lega le misure dei cateti a quella dell'ipotenusa di un triangolo rettangolo:

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

dove:  $a$ ,  $b$  – sono i cateti del triangolo.

$c$  – è l'ipotenusa del triangolo.

$d$  – non c'entra niente.

## 6.6 Scatole

L<sup>A</sup>T<sub>E</sub>X costruisce le sue pagine racchiudendo tutto in scatole. Dapprima, ciascuna lettera è una piccola scatola, che viene poi incollata ad altre lettere per formare parole. Queste ultime sono a loro volta incollate ad altre parole, ma con una colla speciale, che è elastica perché una serie di parole possa essere ristretta o espansa per riempire esattamente una linea sulla pagina.

Certo, questa è una versione molto semplicistica di ciò che realmente accade, ma il punto è che T<sub>E</sub>X funziona in base a colla e scatole. Le lettere non sono le uniche cose che possono essere inscatolate. Si può mettere praticamente tutto in una scatola, comprese altre scatole. Ogni scatola sarà gestita da L<sup>A</sup>T<sub>E</sub>X come se fosse una singola lettera.

Nei capitoli precedenti abbiamo incontrato alcune scatole, benché ciò non sia stato detto esplicitamente. Ad esempio, sia l'ambiente `tabular`

sia `\includegraphics` producono entrambi una scatola. Ciò significa che è possibile collocare con facilità due tabelle o immagini affiancate. Basta solo assicurarsi che la loro larghezza complessiva non sia maggiore della larghezza del testo.

È possibile anche inserire un capoverso a piacere in una scatola col comando:

```
\parbox[⟨pos⟩]{⟨larghezza⟩}{⟨testo⟩}
```

oppure con l'ambiente:

```
\begin{minipage}[⟨pos⟩]{⟨larghezza⟩}
⟨testo⟩
\end{minipage}
```

Il parametro `⟨pos⟩` può assumere una delle lettere `c`, `t` oppure `b` per controllare l'allineamento verticale della scatola, relativo alla linea di base del testo circostante. L'argomento `⟨larghezza⟩` prende un valore che specifica la larghezza della scatola. La differenza principale tra `minipage` e `\parbox` è che non si possono usare tutti i comandi e gli ambienti all'interno di `parbox`, mentre dentro `minipage` si può usare praticamente tutto.

Mentre `\parbox` e `minipage` possono racchiudere anche un intero capoverso, consentendo l'interruzione di linea, e qualsiasi altra cosa valida in un capoverso, esiste anche una classe di comandi di inscatolamento che opera solo su materiale allineato orizzontalmente. Ne conosciamo già un paio: sono `\mbox` e `\makebox`. Questi comandi possono essere annidati l'uno nell'altro, così che si possono ottenere effetti speciali ricorrendo a questi "trucchi". Uno degli usi più noti di `\mbox` consiste nel racchiudere una parola in modo che essa non possa subire una cesura a fine riga.

Nel comando

```
\makebox{⟨larghezza⟩}[⟨pos⟩]{⟨testo⟩}
```

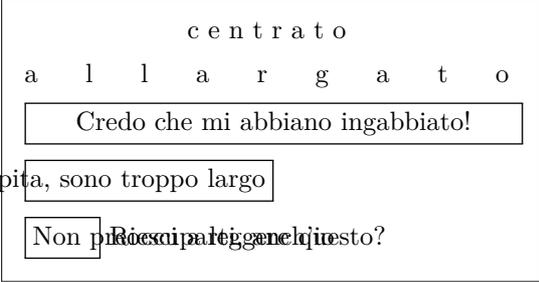
`larghezza` definisce la larghezza della scatola risultante, come vista dall'esterno.<sup>6</sup> Oltre alle espressioni di lunghezza si possono usare anche `\width`, `\height`, `\depth` e `\totalheight` nel parametro `⟨larghezza⟩`. Essi sono impostati dai valori ottenuti misurando il `⟨testo⟩` composto. Il parametro `pos` assume un valore di una lettera: `center` (testo centrato), `flushleft` (testo allineato a sinistra), `flushright` (testo allineato a destra), e infine `spread` (parole spaziate che in linguaggio tipografico diventa "giustificazione forzata") per riempire la scatola.

<sup>6</sup>Ciò significa che essa può essere più piccola del contenuto della scatola. Si può persino impostare la sua larghezza di 0pt, in modo che il testo all'interno della scatola sarà composto con una posizione precisa, ma che può sovrapporsi al testo circostante.

Il comando `\framebox` opera esattamente allo stesso modo di `\makebox`, ma disegna una cornice attorno al testo.

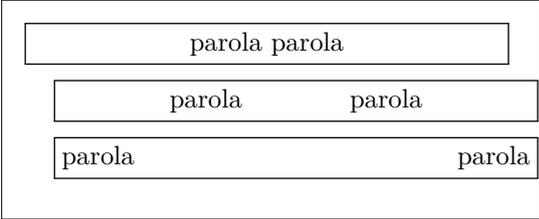
Il seguente esempio mostra alcune delle cose che si possono fare con i comandi `\makebox` e `\framebox`:

```
\parindent=0pt
\makebox[\textwidth]{%
  c e n t r a t o}\par
\makebox[\textwidth][s]{%
  a l l a r g a t o}\par
\framebox[1.26\width][c]{Credo che mi
  abbiano ingabbiato!} \par
\framebox[0.8\width][r]{Caspita, Caspita, sono troppo largo} \par
\framebox[1cm][l]{Non preoccuparti, Non preoccuparti, non preoccuparti,
  anch'io}
Riesci a leggere questo?
```



Si possono usare nell'argomento anche dei comandi di spaziatura, per esempio `\hfil`, `\hfill`, o `\hfilll`; si tratta di spazi elastici con larghezza naturale nulla e allargamento “infinito”. Al variare del numero delle “l”, la loro spinta di allargamento diventa “più forte”.

```
\framebox[\textwidth][c]{parola
  parola}\par
\framebox[\textwidth][c]{parola\hfil
  parola}\par
\framebox[\textwidth][c]{parola\hfill
  parola}
```



Ora che sappiamo controllare l'orizzontale, ovviamente il prossimo passo è provvedere al controllo del verticale. Nessun problema per L<sup>A</sup>T<sub>E</sub>X. Il comando:

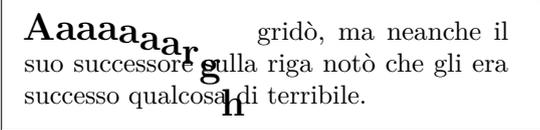
```
\raisebox{<sollevamento>}[<sopra-la-linea-di-base>][<sotto-la-linea-di-base>]{<testo>}
```

consente di definire le proprietà verticali di una scatola. Si può ricorrere a `\width`, `\height`, `\depth` e a `\totalheight` per i primi tre parametri, in modo da manipolare la dimensione della scatola contenuta all'interno dell'argomento *testo*.

```

\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa}\raisebox{-0.2ex}{a}%
\raisebox{-0.6ex}{a}%
\raisebox{-0.9ex}{a}%
\raisebox{-1.4ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}
gridò, ma neanche il suo successore
sulla riga notò che gli era successo
qualcosa di terribile.

```



## 6.7 Linee

Alcune pagine addietro si è fatto ricorso al comando:

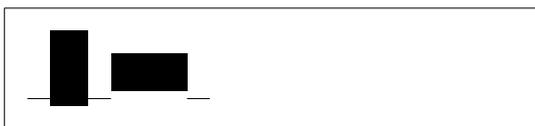
```
\rule[⟨sollevamento⟩]{⟨larghezza⟩}{⟨altezza⟩}
```

Nell'uso normale esso produce una semplice scatola nera.

```

\rule{3mm}{0.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{0.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{0.1pt}

```



Ciò è utile per disegnare linee orizzontali e verticali. La linea sulla pagina del titolo di questo manualetto, ad esempio, è stata creata col comando `\rule`.

Fine



# Appendice A

## Installazione di L<sup>A</sup>T<sub>E</sub>X

Knuth pubblicò i file sorgente di T<sub>E</sub>X molti anni fa, quando nessuno conosceva né Open Source, né Free Software. La licenza legata a T<sub>E</sub>X permette di fare quello che si vuole con quei sorgenti, ma il risultato si può chiamare T<sub>E</sub>X solo se il programma passa una serie di controlli che Knuth ha creato. Questo ha portato al fatto che oggi abbiamo alcune implementazioni libere disponibili per qualunque sistema operativo. Questa appendice dà alcune indicazioni per far funzionare T<sub>E</sub>X sotto Linux, Mac e Windows.

### A.1 Che cosa installare

Per usare L<sup>A</sup>T<sub>E</sub>X su qualunque calcolatore bisogna disporre di diversi programmi.

1. I programmi T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X per elaborare i file sorgente al fine di ottenere i file di uscita in formato PDF o DVI.
2. Un editor di testi per comporre i file sorgente L<sup>A</sup>T<sub>E</sub>X. Alcuni editor permettono di lanciare direttamente il compilatore L<sup>A</sup>T<sub>E</sub>X.
3. Un visualizzatore di file in formato PDF o DVI per analizzare le anteprime e per stampare i documenti. Spesso l'editor include direttamente questa funzionalità.

### A.2 Editor multiplatforma

Mentre T<sub>E</sub>X è disponibile per molte piattaforme differenti, gli editor sono generalmente specifici per ogni sistema operativo.

L'editor multiplatforma Texmaker piace a molti utenti. Senza contare che si tratta di un editor molto utile con il visualizzatore PDF integrato e la evidenziazione della sintassi, ha il vantaggio di funzionare nello stesso modo

su Windows, Mac e Linux. Si veda <http://www.xmlmath.net/texmaker> per ulteriori informazioni. Esiste anche un “fork” di Texmaker che si chiama TeXstudio nel sito <http://texstudio.sourceforge.net/>. Anche questo è costantemente aggiornato per le tre piattaforme maggiori; molti lo preferiscono rispetto a Texmaker.

Le distribuzioni recenti del sistema  $\text{\TeX}$ , come già detto, contengono anche l’editor  $\text{\TeX}$ works (in mancanza si veda <http://texworks.org/>) che è multipiattaforma, ed è basato su  $\text{\TeX}$ Shop. Siccome  $\text{\TeX}$ works è basato sulle librerie Qt, è disponibile per ogni piattaforma che ne faccia uso: (Mac, Windows, Linux).

Qui di seguito ci sono informazioni per l’installazione del sistema  $\text{\TeX}$  per alcuni editor specifici di ciascun sistema operativo.

### A.3 $\text{\TeX}$ su Mac

Parlando di piattaforme Mac qui ci riferiamo a quelle moderne che dispongono del sistema operativo Max OS X, non per i sistemi operativi come Mac OS 9 o ancora più datati. Ora (ottobre 2020) questo è arrivato alla versione 10.15.7; esso opera su macchine con i processori a 64 bit.

Per l’installazione basta scaricare `MacTeX.pkg`. Si tratta di una versione precompilata specifica per Mac OS X e si installa cliccando il file appena scaricato e seguendo le istruzioni specificate nella finestra di installazione. `MacTeX.pkg` produce una installazione completa del sistema  $\text{\TeX}$  e contiene alcuni altri strumenti.  $\text{\MacTeX}$  per macchine a 64 bit si scarica da <http://www.tug.org/mactex/>.

Con macchine col processore a 32 bit sono disponibili versioni di  $\text{\MacTeX}$  adatte per quel processore, ma bisogna cercarle apposta nel sito <http://www.tug.org/mactex/>.

### A.4 $\text{\TeX}$ su Windows

Per Windows ci sono due possibilità:  $\text{\TeX}$  Live e  $\text{\MiKTeX}$ .

#### $\text{\TeX}$ Live

$\text{\TeX}$  Live è la distribuzione ufficiale, uguale per tutte le piattaforme, anche se gli eseguibili che contiene sono specifici per ciascuna. Viene mantenuto da una squadra di persone che lavorano a diretto contatto con i curatori dell’archivio ufficiale del sistema  $\text{\TeX}$ .

L’installazione di  $\text{\TeX}$  Live si fa scaricando dal sito <http://www.tug.org/texlive/acquire-iso.html> il file ISO, immagine del disco di installazione; si tratta di un file molto grande che può richiedere parecchio tempo

per essere scaricato dalla rete. Sui sistemi operativi Windows più recenti, da Windows 8 in poi, basta cliccare sul nome del file `.iso`, perché si apra come un disco montato sulla macchina. Si clicchi sul nome del file `install-tl-windows.bat` che provvede all'installazione *completa*.

### MiK<sub>T</sub>E<sub>X</sub>

MiK<sub>T</sub>E<sub>X</sub> è nato come distribuzione per Windows; recentemente ne sono apparse le versioni anche per Linux e Mac, ma sono ancora sperimentali.

Sebbene se ne possa fare l'installazione completa (consigliata), è anche possibile installarne una versione di base in grado di autoscaricare al volo dalla rete gli eventuali pacchetti mancanti. Questa funzionalità sembra gradita agli utenti di Windows, che, però, per usare questa funzionalità dipendono dalla rete Internet e dai suoi capricci.

Si scarichi MiK<sub>T</sub>E<sub>X</sub> da <http://www.miktex.org/>. Si suggerisce di scaricare e installare la versione completa, che rende totalmente indipendenti dalla rete.

## A.5 T<sub>E</sub>X su Linux

Se si lavora con Linux, esiste la possibilità che il sistema T<sub>E</sub>X sia già installato; altrimenti lo si può installare con il *package manager* specifico di ogni piattaforma Linux. In ogni caso, se non sono già installati, quello che occorre è elencato qui di seguito:

- T<sub>E</sub>X Live – l'installazione preferibile per Linux; scegliere l'installazione completa.
- Emacs (con AUC<sub>T</sub>E<sub>X</sub>) – un editor che si integra finemente con L<sup>A</sup>T<sub>E</sub>X attraverso il plug-in AUC<sub>T</sub>E<sub>X</sub>.
- Ghostscript e gs – un gestore e un visualizzatore di file PostScript e PDF.
- Xpdf – un visualizzatore di file PDF.
- ImageMagick – un programma gratuito per la conversione di formato delle immagini a matrici di punti.
- GIMP – un programma gratuito simile a PhotoShop<sup>TM</sup>.
- Inkscape – un programma gratuito simile a Illustrator<sup>®</sup> o a CorelDraw<sup>®</sup>.

Se si desidera un editor più simile a quelli che girano su Windows, ci sono sempre TeXstudio, Texmaker e TeXworks, i cui nomi, pur contenendo la stringa `Tex` o `TeX`, non contengono il logo T<sub>E</sub>X.

## A.6 Raccomandazioni

Sia T<sub>E</sub>X live e MacT<sub>E</sub>X, sia MiK<sub>T</sub>E<sub>X</sub> vengono aggiornati frequentemente; perciò si raccomanda agli utenti di usare gli strumenti tipici di ciascuna delle tre distribuzioni per aggiornare la propria installazione. Talvolta gli aggiornamenti non riguardano solo l'aggiunta di nuove funzionalità ma anche la correzione di piccoli o grandi bug, che vengono segnalati dagli utenti ai curatori di quelle distribuzioni.

## A.7 L'editor

Generalmente, installando il sistema T<sub>E</sub>X, sia con T<sub>E</sub>X Live, sia con MiK<sub>T</sub>E<sub>X</sub>, viene installato anche un editor, Texmaker o TeXworks. Con le macchine Linux potrebbe essere necessario servirsi di Emacs+AU<sub>C</sub>T<sub>E</sub>X, ma se non piace o se lo si trova troppo difficile, se ne può sempre scaricare un altro dalla rete; anche con MiK<sub>T</sub>E<sub>X</sub> potrebbe essere comodo installare un altro fra gli editor TeXworks, o TeXstudio, o Texmaker.

L'installazione di MacT<sub>E</sub>X comprende anche l'editor specifico T<sub>E</sub>XShop, molto ben integrato con il sistema operativo Mac OS X. Se non piacesse, si può installare anche TeXstudio o Texmaker.

**Anteprime dei documenti in formato PDF** — Gli editor T<sub>E</sub>XShop, TeXworks, TeXstudio e Texmaker contengono il loro visualizzatore PDF integrato e sincronizzato con il file sorgente del documento; 'sincronizzato' significa che si passa dalla finestra di editing al punto corrispondente della finestra di visualizzazione e viceversa con un semplice click.

Per le macchine Windows, se si desidera usare autonomamente un altro visualizzatore PDF, il programma Sumatra PDF (<http://blog.kowalczyk.info/software/sumatrapdf/>) svolge un buon lavoro (solo sulle piattaforme Windows) e può anche venire collegato al programma di editing in modo da avere le stesse funzionalità degli editor forniti di visualizzatori integrati.

**Anteprime dei documenti in formato DVI** — Per l'ormai raro formato DVI (era l'unico formato alla nascita di T<sub>E</sub>X, oggi usato molto raramente) MiK<sub>T</sub>E<sub>X</sub> dovrebbe contenere il visualizzatore Yap; gli editor nominati sopra sono già autosufficienti per il formato PDF, ma alcuni convertono al volo il formato DVI in formato PDF.

## A.8 Lavorare con i file grafici

Per gestire i file grafici in formato vettoriale EPS e PDF può essere utile disporre del programma ghostscript e del suo visualizzatore interattivo GView, o gv su Linux, che si possono scaricare entrambi da [https://www.ghostscript.com/~chrisl/old\\_wisc/](https://www.ghostscript.com/~chrisl/old_wisc/).

Per la gestione delle immagini in formato bitmapped si consiglia il programma GIMP (<https://www.gimp.org/downloads/>); è un programma gratuito che svolge più o meno le stesse funzioni di PhotoShop™.



# Bibliografia

Tutti i file con estensione .pdf citati in questa bibliografia sono consultabili semplicemente con il comando `texdoc` seguito dal nome del file (senza estensione).

- [1] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, seconda edizione, 1994, ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The T<sub>E</sub>Xbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, seconda edizione, 1984, ISBN 0-201-13448-9. Aggiornato al 1996 nella Millenium Edition.
- [3] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley. *The L<sup>A</sup>T<sub>E</sub>X Companion*, (2nd Edition). Addison-Wesley, Reading, Massachusetts, 2004, ISBN 0-201-36299-6.
- [4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.
- [5] Installando un versione completa e aggiornata del sistema T<sub>E</sub>X non esiste una “Local Guide”, ma è tutto documentato e leggibile dando da linea di comando il comando `texdoc` seguito dal nome del documento che si vuole leggere; per T<sub>E</sub>X Live, potrebbe essere `texdoc texlive`, ma l’installazione contiene molti altri documenti di carattere generale come questo. Nelle installazioni centralizzate su grandi elaboratori, potrebbe non essere disponibile una “Local Guide”; in questi casi bisogna rivolgersi al personale del centro di calcolo.
- [6] L<sup>A</sup>T<sub>E</sub>X3 Project Team. *L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> for authors*. Distribuito con T<sub>E</sub>X Live col nome `usrguide.pdf`.
- [7] L<sup>A</sup>T<sub>E</sub>X3 Project Team. *L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> for Class and Package writers*. Distribuito con T<sub>E</sub>X Live col nome `clsguide.pdf`.
- [8] L<sup>A</sup>T<sub>E</sub>X3 Project Team. *L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> Font selection*. Distribuito con T<sub>E</sub>X Live col nome `fntguide.pdf`.

- [9] David P. Carlisle. *Packages in the ‘graphics’ bundle*. Distribuito con T<sub>E</sub>X Live col nome `grfguide.pdf`.
- [10] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of L<sup>A</sup>T<sub>E</sub>X’s verbatim Environments*. Distribuito con T<sub>E</sub>X Live col nome `verbatim.pdf`.
- [11] Vladimir Volovich, Werner Lemberg and L<sup>A</sup>T<sub>E</sub>X3 Project Team. *Cyrillic languages support in L<sup>A</sup>T<sub>E</sub>X*. Distribuito con T<sub>E</sub>X Live col nome `cyrguide.pdf`.
- [12] Graham Williams. *The TeX Catalogue* è una lista molto ricca contenente i pacchetti legati a T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X. Disponibile su [CTAN://help/Catalogue/catalogue.html](http://CTAN://help/Catalogue/catalogue.html).
- [13] Kristoffer H. Rose. *Xy-pic User’s Guide*. Distribuito con T<sub>E</sub>X Live col nome `xyguide.pdf`.
- [14] John D. Hobby. *A User’s Manual for METAPOST*. Distribuito con T<sub>E</sub>X Live col nome `mpman.pdf`.
- [15] Alan Hoenig. *T<sub>E</sub>X Unbound*. Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)
- [16] Urs Oswald. *Graphics in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*, contiene alcuni file in linguaggio Java per generare cerchi ed ellissi arbitrarie nell’ambiente `picture` e un testo introduttivo a METAPOST: *METAPOST A tutorial*. Entrambi possono essere scaricati da <http://www.ursoswald.ch>.
- [17] Till Tantau. *TikZ & PGF Manual*. Distribuito con T<sub>E</sub>X Live col nome `pgfmanual.pdf`
- [18] François Charette. *Polyglossia: A Babel Replacement for X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X*. Distribuito con T<sub>E</sub>X Live col nome `polyglossia.pdf`.
- [19] François Charette. *An ArabT<sub>E</sub>X-like interface for typesetting languages in Arabic script with X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X*. Distribuito con T<sub>E</sub>X Live col nome `arabxetex.pdf`.
- [20] Will Robertson and Khaled Hosny. *The fontspec package*. Distribuito con T<sub>E</sub>X Live col nome `fontspec.pdf`.
- [21] Apostolos Syropoulos. *The xgreek package*. Distribuito con T<sub>E</sub>X Live col nome `xgreek.pdf`.
- [22] Vafa Khalighi. *The bidi package*. Distribuito con T<sub>E</sub>X Live col nome `bidi.pdf`.

- 
- [23] Vafa Khalighi. *The XePersian package*. Distribuito con T<sub>E</sub>X Live col nome `xepersian-doc.pdf`.
- [24] Wenchang Sun. *The xeCJK package*. Distribuito con T<sub>E</sub>X Live col nome `xeCJK.pdf`.

# Riferimenti italiani

Questa bibliografia italiana contiene un piccolo numero delle guide pubblicate dai membri presenti e passati del Gruppo degli Utenti Italiani di T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X (G<sub>U</sub>I<sub>T</sub>). La sezione Documentazione del sito [www.guitex.org](http://www.guitex.org) contiene i link per scaricare molte altre guide. Il lettore è invitato a tenere conto di tutta questa vasta documentazione. Non trascuri nemmeno gli articoli pubblicati sulla rivista del G<sub>U</sub>I<sub>T</sub>, *ArsT<sub>E</sub>Xnica*, reperibili nella sezione ArsT<sub>E</sub>Xnica: <https://www.guitex.org/home/en/arstexnica>.

- [1] C. Beccari, *Il L<sup>A</sup>T<sub>E</sub>X Reference Manual commentato*, <http://www.guitex.org/home/images/doc/GuideGuIT/latexhandbookcommentato.pdf>
- [2] C. Beccari, *Introduzione all'arte della composizione tipografica con L<sup>A</sup>T<sub>E</sub>X*, <http://www.guitex.org/home/images/doc/guidaguit-b5.pdf>
- [3] C. Beccari, *Font e tipografia*, <http://www.guitex.org/home/images/doc/GuideGuIT/guidafont.pdf>
- [4] C. Beccari, *Tipocomporre in italiano*, <http://www.guitex.org/home/images/doc/GuideGuIT/ComporreItaliano.pdf>
- [5] C. Beccari, *Regole e consigli per comporre la matematica delle scienze sperimentali*, <http://www.guitex.org/home/images/doc/GuideGuIT/ComporreMatematica.pdf>
- [6] C. Beccari e T. Gordini, *L'arte di scrivere in diverse lingue con {Xe/Lua}L<sup>A</sup>T<sub>E</sub>X*, <http://www.guitex.org/home/images/doc/ArteLingue.pdf>
- [7] A. De Marco, *Scrivere la tesi di laurea in L<sup>A</sup>T<sub>E</sub>X*, <http://www.guitex.org/home/images/doc/GuideGuIT/IntroTesi.pdf>
- [8] E. Gregorio, *Appunti di programmazione in T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X*, <http://profs.sci.univr.it/~gregorio/introtex.pdf>
- [9] L. Pantieri e T. Gordini, *L'Arte di scrivere con L<sup>A</sup>T<sub>E</sub>X*, [http://www.lorenzopantieri.net/LaTeX\\_files/ArteLaTeX.pdf](http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf)

- 
- [10] L. Pantieri, *L<sup>A</sup>T<sub>E</sub>X per l'impaziente*, [http://www.lorenzopantieri.net/LaTeX\\_files/LaTeXimpaziente.pdf](http://www.lorenzopantieri.net/LaTeX_files/LaTeXimpaziente.pdf)
- [11] L. Pantieri, *L<sup>A</sup>T<sub>E</sub>Xpedia*, [http://www.lorenzopantieri.net/LaTeX\\_files/LaTeXpedia.pdf](http://www.lorenzopantieri.net/LaTeX_files/LaTeXpedia.pdf)
- [12] F. Vomiero, *Comporre la bibliografia in L<sup>A</sup>T<sub>E</sub>X: bibl<sub>at</sub>ex e i software di gestione tipografica*, <http://www.guitex.org/home/images/doc/GuideGuIT/bibliografia.pdf>

# Indice analitico

## Simboli

- `\!`, 64
- `"`, 22
- `$`, 48
- `"`, 22
- `\,`, 50, 63, 64
- `-`, 23
- `—`, 23
- `\-`, 21
- `-`, 23
- `—`, 23
- `..`, space after, 30
- `...`, 25
- `\/`, 25
- `\:`, 63
- `\;`, 63
- `\@`, 30
- `\[`, 49
- `\`, 19, 35, 36, 39
- `\*`, 19
- `\]`, 49
- `"`, 22
- `~`, 30
  
- A**
- A4 paper, 11
- A5 paper, 11
- å, 27
- abstract, 37
- accenti, 26
  - matematici, 52
- acuto, 27
- `\addtolength`, 131
  
- æ, 27
- align, 58, 60, 61
- align\*, 61
- alignat, 61
- alignat\*, 61
- aligned, 60, 61
- alignn, 69
- allineati a destra, 35
- allineati a sinistra, 35
- `\Alpha`, 71
- `\alpha`, 51
- ambienti, 34
  - Bmatrix, 62
  - Vmatrix, 63
- abstract, 37
- align\*, 61
- alignat\*, 61
- alignat, 61
- aligned, 60, 61
- alignn, 69
- align, 58, 60, 61
- amsmath, 60
- array, 61–63
- block, 96
- bmatrix, 62
- cases, 62
- center, 35
- comment, 7
- description, 35
- displaymath, 48
- enumerate, 35
- eqnarray, 57–60
- equation\*, 48, 49, 57

- equation, 48, 49, 57, 58  
 falign\*, 61  
 falign, 61  
 figure, 35, 44, 45  
 flushleft, 35, 36  
 flushright, 35  
 frame, 96  
 gather\*, 61  
 gathered, 61  
 gather, 61  
 german, 29  
 itemize, 35  
 lscommand, 117  
 matrix, 62  
 minipage, 133  
 multiline\*, 57  
 multiline, 57, 58  
 parbox, 133  
 pict2e, 98, 112  
 picture, 97–99, 109, 111, 113, 144  
 pmatrix, 62  
 proof, 59, 68  
 quotation, 36  
 quote, 36  
 sloppypar, 20  
 split, 61  
 subequations, 60  
 table, 44, 45  
 tabular, 38, 39, 132  
 thebibliography, 79  
 tikzpicture, 113  
 verbatim, 83  
 verse, 36  
 vmatrix, 63  
 amsbsy, 66  
 amsfonts, 50, 51, 70  
 amsmath, 47–50, 53–55, 57, 58, 60–64, 66, 67, 69  
 amsmath, 60  
 amssymb, 50, 65, 70, 75–78  
 amssymn, 51  
 amsthm, 67, 68  
 \and, 32  
 Anteprema.app, 42  
 apice, 51  
 apostrofo, 52  
 \appendix, 31, 33  
 \arccos, 53  
 \arcsin, 53  
 \arctan, 53  
 \arg, 53  
 argomento, 6  
 aritmetica modulare, 54  
 array, 61–63  
 \arraystretch, 41  
 \author, 32, 88, 96  
 avanzamento di riga, 125  
**B**  
 B5 paper, 11  
 babel, 21, 27, 28  
 \backmatter, 33  
 \bar, 52  
 barra inversa, 5  
 beamer, 93, 94, 96, 98  
 \begin, 7, 28, 34, 79, 96, 99  
 \Beta, 71  
 \beta, 51  
 \beveljoin, 106  
 \bibitem, 79  
 biblatex, 80  
 \Big, 56  
 \big, 56  
 \Bigg, 56  
 \bigg, 56  
 \bigskip, 129  
 \binomial, 54  
 blackboard bold, 50  
 block, 96  
 bm, 66  
 \bm, 66  
 Bmatrix, 62  
 bmatrix, 62  
 \bmod, 54  
 bold symbols, 50  
 \boldmath, 66  
 \boldsymbol, 66

- booktabs, 39
- \buttcap, 106
- C
- calc, 131
- capoverso, 17
- \caption, 45
- caratteri riservati, 5
- caratteri speciali, 26
- cases, 62
- \cbezier, 100
- \cdot, 52
- \cdots, 52
- center, 35
- \centering, 35
- \chapter, 31, 32, 82
- \chaptermark, 82
- \ci, 117
- \circle, 104, 111
- \circle\*, 104
- \cite, 79
- classe article, 10
- classe beamer, 10
- classe book, 10
- classe minimal, 10
- classe proc, 10
- classe report, 10
- classe slides, 10
- \cleardoublepage, 46
- \clearpage, 46
- \cline, 39
- codifica dei font, 13
- coefficiente binomiale, 54
- color, 102
- comandi, 5
  - \!, 64
  - \,, 50, 63, 64
  - \-, 21
  - \/, 25
  - \:, 63
  - \;, 63
  - \@, 30
  - \[, 49
  - \\, 19, 35, 36, 39
- \\\*, 19
- \], 49
- \addtolength, 131
- \Alpha, 71
- \alpha, 51
- \and, 32
- \appendix, 31, 33
- \arccos, 53
- \arcsin, 53
- \arctan, 53
- \arg, 53
- \arraystretch, 41
- \author, 32, 88, 96
- \backmatter, 33
- \bar, 52
- \begin, 7, 28, 34, 79, 96, 99
- \Beta, 71
- \beta, 51
- \beveljoin, 106
- \bibitem, 79
- \Big, 56
- \big, 56
- \Bigg, 56
- \bigg, 56
- \bigskip, 129
- \binomial, 54
- \bm, 66
- \bmod, 54
- \boldmath, 66
- \boldsymbol, 66
- \buttcap, 106
- \caption, 45
- \cbezier, 100
- \cdot, 52
- \cdots, 52
- \centering, 35
- \chapter, 31, 32, 82
- \chaptermark, 82
- \ci, 117
- \circle, 104, 111
- \circle\*, 104
- \cite, 79
- \cleardoublepage, 46
- \clearpage, 46

- `\cline`, 39
- `\cos`, 53
- `\cosh`, 53
- `\cot`, 53
- `\coth`, 53
- `\csc`, 53
- `\date`, 32
- `\ddots`, 52
- `\DeclareMathOperator`, 53
- `\deg`, 53
- `\Delta`, 51
- `\depth`, 133, 134
- `\det`, 53
- `\dfrac`, 54
- `\dim`, 53
- `\displaystyle`, 65
- `\documentclass`, 7, 9, 10, 14, 20
- `\dum`, 117
- `\emph`, 34, 125
- `\end`, 7, 28, 34, 79, 99
- `\eqref`, 48
- `\exp`, 53
- `\fbox`, 22
- `\foldera`, 110
- `\folderb`, 110
- `\footnote`, 33
- `\footskip`, 130
- `\foreignlanguage`, 28
- `\foreignlanguage*`, 28
- `\frac`, 54
- `\framebox`, 107, 134
- `\frametitle`, 96
- `\frenchspacing`, 30
- `\frontmatter`, 32
- `\fussy`, 20
- `\Gamma`, 51
- `\gamma`, 51
- `\gcd`, 53
- `\greektxt`, 30
- `\hat`, 52
- `\headheight`, 130
- `\headsep`, 130
- `\height`, 133, 134
- `\hfil`, 134
- `\hfill`, 134
- `\hfilll`, 134
- `\hline`, 39
- `\hom`, 53
- `\href`, 88
- `\hspace`, 120, 127
- `\hspace*`, 127
- `\hypersetup`, 86
- `\hyphenation`, 21
- `\idotsint`, 64
- `\ignorespacesafterend`, 120
- `\ignorespaces`, 120
- `\iiint`, 64
- `\iint`, 64
- `\include`, 15
- `\includegraphics`, 42, 43, 133
- `\includeonly`, 15
- `\indent`, 127
- `\index`, 81, 82
- `\inf`, 53
- `\input`, 16
- `\institute`, 96
- `\int`, 55
- `\item`, 35
- `\ker`, 53
- `\label`, 33, 45, 48
- `\LaTeX`, 22
- `\LaTeXe`, 22
- `\ldots`, 25, 52
- `\left`, 55, 56, 62
- `\lefteqn`, 59
- `\leftmark`, 82
- `\lg`, 53
- `\lim`, 53
- `\liminf`, 53
- `\limsup`, 53
- `\Line`, 102
- `\line`, 100, 103
- `\linebreak`, 20
- `\linespread`, 125, 126
- `\linethickness`, 105, 108
- `\listoffigures`, 31, 45

- `\listoftables`, 31, 45
- `\ln`, 53
- `\log`, 53
- `\mainmatter`, 32
- `\makebox`, 107, 109, 133, 134
- `\makeindex`, 81
- `\maketitle`, 32
- `\marginparpush`, 130
- `\marginparsep`, 130
- `\marginparwidth`, 130
- `\mathbb`, 50
- `\mathbf`, 66
- `\max`, 53
- `\mbox`, 22, 133
- `\medskip`, 129
- `\min`, 53
- `\miterjoin`, 106
- `\mkern`, 58, 128
- `\multicolumn`, 40
- `\multipt`, 99, 108, 109
- `\newcommand`, 64, 118, 119
- `\newenvironment`, 119
- `\newline`, 19
- `\newpage`, 20
- `\newsavebox`, 109
- `\newtheorem`, 66, 67
- `\noindent`, 127
- `\nolinebreak`, 20
- `\nonumber`, 58
- `\nopagebreak`, 20
- `\not`, 72
- `\oddsidemargin`, 130
- `\oval`, 109
- `\overbrace`, 52
- `\overleftarrow`, 53
- `\overline`, 52
- `\overrightarrow`, 53
- `\pagebreak`, 20
- `\pageref`, 33, 85
- `\pagestyle`, 12
- `\paperheight`, 130
- `\paperwidth`, 130
- `\par`, 124
- `\paragraph`, 31
- `\parbox`, 133
- `\parindent`, 126
- `\parskip`, 126
- `\part`, 31
- `\partial`, 54
- `\pause`, 96
- `\phantom`, 65
- `\pmod`, 54
- `\polygon`, 102
- `\polygon*`, 102
- `\polyline`, 102
- `\Pr`, 53
- `\printindex`, 82
- `\prod`, 55
- `\providecommand`, 119
- `\ProvidesPackage`, 121
- `\put`, 99, 100, 104, 109
- `\q bezier`, 97, 100
- `\qedhere`, 68, 69
- `\qqquad`, 50, 64
- `\quad`, 50, 58, 64
- `\raggedleft`, 35
- `\raggedright`, 35
- `\raisebox`, 134
- `\ref`, 33, 45, 85
- `\renewcommand`, 119
- `\renewenvironment`, 119
- `\right`, 55, 56, 62
- `\rightmark`, 82
- `\roundcap`, 106
- `\roundjoin`, 106
- `\rule`, 41, 120, 135
- `\russiantxt`, 30
- `\savebox`, 109
- `\scriptscriptstyle`, 65
- `\scriptstyle`, 65
- `\sec`, 53
- `\section`, 31, 82, 96
- `\sectionmark`, 82
- `\selectfont`, 126
- `\selectlanguage`, 28
- `\setlength`, 99, 126, 131
- `\settodepth`, 131
- `\settoheight`, 131

- `\settowidth`, 131
- `\sin`, 53
- `\sinh`, 53
- `\slash`, 23
- `\sloppy`, 20
- `\smallskip`, 129
- `\smash`, 50
- `\sqrt`, 51
- `\squarecap`, 106
- `\stackrel`, 55
- `\stretch`, 120, 127, 128
- `\subparagraph`, 31
- `\subsection`, 31, 96
- `\subsetionmark`, 82
- `\substack`, 55
- `\subsubsection`, 31
- `\sum`, 55
- `\sup`, 53
- `\surd`, 51
- `\tabcolsep`, 41
- `\tableofcontents`, 31
- `\tag`, 48
- `\tan`, 53
- `\tanh`, 53
- `\TB`, 80
- `\TeX`, 22
- `\textorpdfstring`, 90
- `\text`, 50, 62
- `\textasciitilde`, 23
- `\textbackslash`, 5
- `\textbf`, 81, 90
- `\textcelsius`, 24
- `\textdegree`, 24
- `\texteuro`, 24
- `\textheight`, 130
- `\textit`, 81, 90
- `\textsl`, 81
- `\textstyle`, 65
- `\textwidth`, 130
- `\tfrac`, 54
- `\the`, 132
- `\theoremstyle`, 67
- `\thicklines`, 104, 105
- `\thinlines`, 104, 105
- `\thispagestyle`, 12
- `\title`, 32, 96
- `\titlegraphic`, 96
- `\tnss`, 118
- `\today`, 22
- `\topmargin`, 130
- `\totalheight`, 133, 134
- `\ud`, 64
- `\underbrace`, 52
- `\underline`, 34, 52
- `\unitlength`, 99
- `\usebox`, 109
- `\usepackage`, 10, 14, 24, 25, 47, 81, 86, 98, 120
- `\vdots`, 52
- `\vec`, 53
- `\vector`, 103, 104
- `\verb`, 37, 38
- `\verbatiminput`, 83
- `\vline`, 39
- `\vspace`, 128
- `\vspace*`, 128
- `\widehat`, 52
- `\widetilde`, 52
- `\width`, 133, 134
- comment, 7
- commenti, 6
- corsivo, 122
- `\cos`, 53
- `\cosh`, 53
- `\cot`, 53
- `\coth`, 53
- `\csc`, 53
- csquotes, 28
- D**
- `\date`, 32
- dcolumn, 40
- `\ddots`, 52
- decimal alignment, 40
- `\DeclareMathOperator`, 53
- `\deg`, 53
- `\Delta`, 51
- `\depth`, 133, 134

- derivata parziale, 54
- description, 35
- `\det`, 53
- `\dfrac`, 54
- dieresi, 27
- `\dim`, 53
- dimensione del foglio, 129
- dimensione del font del documento, 11
- dimensione di base del font, 11
- dimensioni, 127
- `displaymath`, 48
- `\displaystyle`, 65
- doc, 13
- `\documentclass`, 7, 9, 10, 14, 20
- dotless i and j, 27
- double line spacing, 125
- dritto, 122
- `\dum`, 117
- E**
- ellissi, 25
- `\emph`, 34, 125
- empty, 12
- `\end`, 7, 28, 34, 79, 99
- enumerate, 35
- `eqnarray`, 57–60
- `\eqref`, 48
- equation
  - L<sup>A</sup>T<sub>E</sub>X, 49
  - amsmath, 49
  - equation, 48, 49, 57, 58
  - equation\*, 48, 49, 57
  - equazioni
    - multiple, 57
  - equazioni lunghe, 56
  - esponente, 51
  - estensione, 12
    - .aux, 15
    - .cls, 14
    - .dtx, 14
    - .dvi, 14
    - .fd, 14
    - .idx, 15, 81
    - .ilg, 15
    - .ind, 15, 82
    - .ins, 14
    - .lof, 15
    - .log, 14
    - .lot, 15
    - .pdf, 14
    - .sty, 14, 84
    - .tex, 9, 14
    - .toc, 15
  - eurosym, 25
  - executive paper, 11
  - `\exp`, 53
  - exscale, 13
  - F**
  - falign, 61
  - falign\*, 61
  - fancyhdr, 82, 83
  - fancyvrb, 84
  - `\fbox`, 22
  - figure, 35, 44, 45
  - file di input, 7
  - flushleft, 35, 36
  - flushright, 35
  - `\foldera`, 110
  - `\folderb`, 110
  - font, 121
    - `\emph`, 122
    - `\footnotesize`, 122, 123
    - `\Huge`, 122, 123
    - `\huge`, 122, 123
    - `\LARGE`, 122, 123
    - `\Large`, 122, 123
    - `\large`, 122, 123
    - `\mathbf`, 123
    - `\mathcal`, 123
    - `\mathit`, 123
    - `\mathnormal`, 123
    - `\mathrm`, 123
    - `\mathsf`, 123
    - `\mathtt`, 123
    - `\normalsize`, 122, 123
    - `\scriptsize`, 122, 123

- `\small`, 122, 123
- `\textbf`, 122
- `\textit`, 122
- `\textmd`, 122
- `\textnormal`, 122
- `\textrm`, 122
- `\textsc`, 122
- `\textsf`, 122
- `\textsl`, 122
- `\texttt`, 122
- `\textup`, 122
- `\tiny`, 122, 123
- font size, 121, 122
- fontenc, 13
- fontspec, 27, 29, 91, 92
- `\footnote`, 33
- `\footskip`, 130
- `\foreignlanguage`, 28
- `\foreignlanguage*`, 28
- formule matematiche, 48
- `\frac`, 54
- frame, 96
- `\framebox`, 107, 134
- `\frametitle`, 96
- frazione, 54
- `\frenchspacing`, 30
- `\frontmatter`, 32
- funzioni
  - matematiche, 53
- funzioni discontinue, 62
- fuori testo, 49
- `\fussy`, 20
- G**
- `\Gamma`, 51
- `\gamma`, 51
- gather, 61
- gather\*, 61
- gathered, 61
- `\gcd`, 53
- geometry, 84
- german, 29
- ghostscript, 140
- graffa
  - orizzontale, 52
  - graffe orizzontali, 52
- grafica, 10
- graphicx, 42, 93
- grassetto, 122
- grave, 27
- `\greektext`, 30
- grouping, 122
- gv, 140
- GView, 140
- H**
- `\hat`, 52
- `\headheight`, 130
- textttheadings, 12
- `\headsep`, 130
- `\height`, 133, 134
- `\hfil`, 134
- `\hfill`, 134
- `\hfilll`, 134
- `\hline`, 39
- `\hom`, 53
- `\href`, 88
- `\hspace`, 120, 127
- `\hspace*`, 127
- hyperref, 86, 89, 90, 93
- `\hypersetup`, 86
- hyphenat, 84
- `\hyphenation`, 21
- I**
- `\idotsint`, 64
- ifthen, 13
- `\ignorespacesafterend`, 120
- `\ignorespaces`, 120
- `\iiint`, 64
- `\iint`, 64
- imakeidx, 13, 80, 82
- inclinato, 122
- `\include`, 15
- `\includegraphics`, 42, 43, 133
- `\includeonly`, 15
- `\indent`, 127
- indentfirst, 127

- `\index`, 81, 82
  - indice, 31
- `\inf`, 53
- `\input`, 16
  - inputenc, 13, 21
- `\institute`, 96
- `\int`, 55
  - internazionale, 26
  - interruzioni di riga, 19
  - ipertesto, 85
- `\item`, 35
  - itemize, 35
- K**
- `\ker`, 53
  - Knuth, Donald E., 1
- L**
- `\label`, 33, 45, 48
  - Lamport, Leslie, 2
- `\LaTeX`, 22
- `\LaTeXe`, 22
  - latexsym, 13
  - layout, 129
  - layout della pagina, 130
  - layouts, 130
- `\ldots`, 25, 52
- `\left`, 55, 56, 62
- `\lefteqn`, 59
- `\leftmark`, 82
  - legal paper, 11
  - legature, 25
  - letter paper, 11
  - lettere greche, 51
  - lettere scandinave, 27
- `\lg`, 53
- `\lim`, 53
- `\liminf`, 53
- `\limsup`, 53
- `\Line`, 102
- `\line`, 100, 103
  - linea
    - orizzontale, 52
- `\linebreak`, 20
  - lineetta, 23
- `\linespread`, 125, 126
- `\linethickness`, 105, 108
  - lingue, 26
  - listings, 84
- `\listoffigures`, 31, 45
- `\listoftables`, 31, 45
- `\ln`, 53
- `\log`, 53
  - longtable, 41
  - lscommand, 117
  - LuaL<sup>A</sup>T<sub>E</sub>X, 93
  - LXfonts, 93
- M**
- MacTeX.pkg, 138
- `\mainmatter`, 32
  - maiuscoletto, 122
- `\makebox`, 107, 109, 133, 134
  - makeidx, 13, 81
  - makeidx package, 80
- `\makeindex`, 81
  - makeindex program, 80
- `\maketitle`, 32
  - margin, 129
- `\marginparpush`, 130
- `\marginparsep`, 130
- `\marginparwidth`, 130
- matematico
  - delimitatore, 56
- `\mathbb`, 50
- `\mathbf`, 66
- mathrsfs, 75
- matrix, 62
- `\max`, 53
- `\mbox`, 22, 133
- `\medskip`, 129
  - meno, 23
    - matematico, 23
- mhchem, 65
- microtype, 93
- MiK<sub>T</sub>E<sub>X</sub>, 139
- `\min`, 53
- minipage, 133

- `\miterjoin`, 106
- `\mkern`, 58, 128
- `\multicolumn`, 40
- `\multipt`, 99, 108, 109
- `multiline`, 57, 58
- `multiline*`, 57
  
- N**
- neretto da lavagna, 50
- `\newcommand`, 64, 118, 119
- `\newenvironment`, 119
- `\newline`, 19
- `\newpage`, 20
- `\newsavebox`, 109
- `\newtheorem`, 66, 67
- `\noindent`, 127
- `\nolinebreak`, 20
- `\nonumber`, 58
- `\nopagebreak`, 20
- `\not`, 72
- `ntheorem`, 68
  
- O**
- `\oddsidemargin`, 130
- œ, 27
- oggetti flottanti, 43
- one sided, 11
- onecolumn, 11
- opzioni, 9
- orizzontale
  - graffa, 52
  - linea, 52
- `\oval`, 109
- `\overbrace`, 52
- overflow hbox, 20
- `\overleftarrow`, 53
- `\overline`, 52
- `\overrightarrow`, 53
  
- P**
- pacchetti, 7
  - amsbsy, 66
  - amsfonts, 50, 51, 70
  - amsmath, 47–50, 53–55, 57, 58, 60–64, 66, 67, 69
  - amssymb, 50, 65, 70, 75–78
  - amssymn, 51
  - amsthm, 67, 68
  - babel, 21, 27, 28
  - beamer, 93, 94, 96, 98
  - biblatex, 80
  - bm, 66
  - booktabs, 39
  - calc, 131
  - color, 102
  - csquotes, 28
  - dcolumn, 40
  - doc, 13
  - eurosym, 25
  - exscale, 13
  - fancyhdr, 82, 83
  - fancyvrb, 84
  - fontenc, 13
  - fontspec, 27, 29, 91, 92
  - geometry, 84
  - graphicx, 42, 93
  - hyperref, 86, 89, 90, 93
  - hyphenat, 84
  - ifthen, 13
  - imakeidx, 13, 80, 82
  - indentfirst, 127
  - inputenc, 13, 21
  - latexsym, 13
  - layout, 129
  - layouts, 130
  - listings, 84
  - longtable, 41
  - LXfonts, 93
  - makeidx, 13, 81
  - mathrsfs, 75
  - mhchem, 65
  - microtype, 93
  - ntheorem, 68
  - pgf, 98, 113, 115
  - pgfplots, 98, 115
  - pict2e, 98–100, 102–105, 108, 109
  - polyglossia, 21, 27–29
  - pstricks, 112

- siunitx, 40
  - syntonly, 13, 16
  - textcomp, 24
  - TikZ, 98
  - tikz, 112, 113
  - verbatim, 7, 83, 84
  - xcolor, 93, 102
  - xfp, 131
  - pacchetto, 10, 117
  - page style
    - empty, 12
    - headings, 12
    - plain, 12
  - \pagebreak, 20
  - \pageref, 33, 85
  - \pagestyle, 12
    - paper size, 11
  - \paperheight, 130
  - \paperwidth, 130
  - \par, 124
  - \paragraph, 31
  - \parbox, 133
  - parbox, 133
  - parentesi graffe, 6, 122
  - parentesi quadre, 6
  - \parindent, 126
    - parola, 82
  - \parskip, 126
  - \part, 31
  - \partial, 54
  - \pause, 96
  - PDF, 85, 90
  - pdfL<sup>A</sup>T<sub>E</sub>X, 93
  - pedice, 51
  - pgf, 98, 113, 115
  - pgfplots, 98, 115
  - \phantom, 65
  - piccole frecce, 52
  - pict2e, 98–100, 102–105, 108, 109
  - pict2e, 98, 112
  - picture, 97–99, 109, 111, 113, 144
  - piegino, 12
  - pilastrino, 41
  - plain, 12
  - pmatrix, 62
  - \pmod, 54
    - polyglossia, 21, 27–29
  - \polygon, 102
  - \polygon\*, 102
  - \polyline, 102
  - POSTSCRIPT, 93, 98
  - \Pr, 53
    - preambolo, 7
    - Preview.app, 42
    - primo, 52
  - \printindex, 82
  - \prod, 55
  - proof, 59, 68
  - \providecommand, 119
  - \ProvidesPackage, 121
  - pstricks, 112
  - puntini
    - diagonali, 52
    - orizzontali, 52
    - tre, 52
    - verticali, 52
  - punto, 25, 52
  - \put, 99, 100, 104, 109
- Q**
- \qbezier, 97, 100
  - \qedhere, 68, 69
  - \qqquad, 50, 64
  - \quad, 50, 58, 64
  - quotation, 36
  - quote, 36
- R**
- radice quadrata, 51
  - \raggedleft, 35
  - \raggedright, 35
  - \raisebox, 134
  - \ref, 33, 45, 85
  - \renewcommand, 119
  - \renewenvironment, 119
    - riferimenti incrociati, 33
  - \right, 55, 56, 62
  - \rightmark, 82

- `\roundcap`, 106
- `\roundjoin`, 106
- `\rule`, 41, 120, 135
- `\russiantxt`, 30
- S**
- `\savebox`, 109
- scartamento, 125
- `\scriptscriptstyle`, 65
- `\scriptstyle`, 65
- `\sec`, 53
  - secondo, 52
- `\section`, 31, 82, 96
- `\sectionmark`, 82
  - segno di integrale, 55
  - segno di prodotto, 55
  - segno di somma, 55
- `\selectfont`, 126
- `\selectlanguage`, 28
  - senza grazie, 122
- `\setlength`, 99, 126, 131
- `\settodepth`, 131
- `\settoheight`, 131
- `\settowidth`, 131
  - simboli neri, 66
  - simbolo di grado, 24
- `\sin`, 53
- `\sinh`, 53
  - siunitx, 40
- Slash, 23
- `\slash`, 23
- `\sloppy`, 20
  - sloppypar, 20
- `\smallskip`, 129
- `\smash`, 50
  - spaziatura, 4
    - modo matematico, 50
- spaziature matematiche, 63
- spazio, 4
  - all'inizio di una riga, 4
  - orizzontale, 127
  - verticale, 128
- spazio bianco
  - dopo i comandi, 6
  - specificatore di posizionamento, 44
- split, 61
- `\sqrt`, 51
- `\squarecap`, 106
- `\stackrel`, 55
  - stile testo, 49
- stili di pagina, 12
- `\stretch`, 120, 127, 128
  - struttura, 7
  - subequations, 60
- `\subparagraph`, 31
- `\subsection`, 31, 96
- `\subsetionmark`, 82
- `\substack`, 55
- `\subsubsection`, 31
- `\sum`, 55
- `\sup`, 53
- `\surd`, 51
  - syntonly, 13, 16
- T**
- `\tabcolsep`, 41
  - tabella, 38
- table, 44, 45
- `\tableofcontents`, 31
  - tabular, 38, 39, 132
- `\tag`, 48
- `\tan`, 53
- `\tanh`, 53
- `\TB`, 80
  - tela, 99
- testatina, 12
- testo a colori, 10
- `\TeX`, 22
- `\texorpdfstring`, 90
- `\text`, 50, 62
- `\textasciitilde`, 23
- `\textbackslash`, 5
- `\textbf`, 81, 90
- `\textcelsius`, 24
  - textcomp, 24
- `\textdegree`, 24
- `\texteuro`, 24
- `\textheight`, 130

- `\textit`, 81, 90
  - `\textsl`, 81
  - `\textstyle`, 65
  - `\textwidth`, 130
  - `\tfrac`, 54
  - `\the`, 132
    - The L<sup>A</sup>T<sub>E</sub>X Project, 2
    - thebibliography, 79
  - `\theoremstyle`, 67
  - `\thicklines`, 104, 105
  - `\thinlines`, 104, 105
  - `\thispagestyle`, 12
  - TikZ, 98
  - tikz, 112, 113
  - tikzpicture, 113
  - tilde, 23
  - tilde (  $\sim$  ), 30
  - tildi larghe, 52
  - tipi di file, 12
  - title, 11
  - `\title`, 32, 96
  - `\titlegraphic`, 96
    - titolo, 32
    - titolo del documento, 11
  - `\tnss`, 118
  - `\today`, 22
    - tondo, 122
  - `\topmargin`, 130
  - `\totalheight`, 133, 134
    - tratti, 23
    - trattino, 23
    - tratto, 23
    - two sided, 11
    - twocolumn, 11
- U**
- `\ud`, 64
  - `\underbrace`, 52
    - underfull hbox, 20
  - `\underline`, 34, 52
  - `\unitlength`, 99
    - units, 128
    - unità di misura, 127
    - URL link, 23
  - `\usebox`, 109
  - `\usepackage`, 10, 14, 24, 25, 47, 81, 86, 98, 120
    - UTF-8, 9
- V**
- `\vdots`, 52
  - `\vec`, 53
  - `\vector`, 103, 104
  - `\verb`, 37, 38
    - verbatim, 7, 83, 84
    - verbatim, 83
  - `\verbatiminput`, 83
    - verse, 36
    - vettori, 53
    - virgola, 25
    - virgolette, 22
  - `\vline`, 39
    - Vmatrix, 63
    - vmatrix, 63
  - `\vspace`, 128
  - `\vspace*`, 128
- W**
- `\widehat`, 52
  - `\widetilde`, 52
  - `\width`, 133, 134
  - WYSIWYG, 2, 3
- X**
- xcolor, 93, 102
  - X<sub>L</sub>L<sup>A</sup>T<sub>E</sub>X, 93
  - X<sub>L</sub>L<sup>A</sup>T<sub>E</sub>X, 90
  - X<sub>E</sub>L<sup>A</sup>T<sub>E</sub>X, 90
  - xfp, 131