

LORENZO PANTIERI & TOMMASO GORDINI

L'ARTE

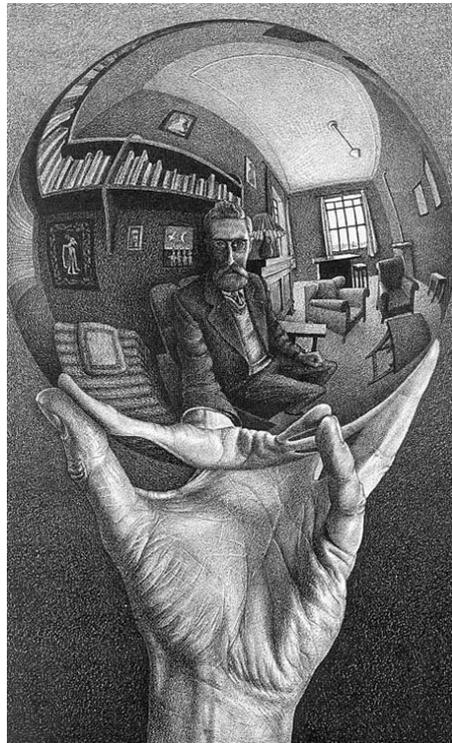
DI SCRIVERE CON

LATEX



LORENZO PANTIERI & TOMMASO GORDINI

*L'*ARTE DI SCRIVERE CON L^AT_EX



EDIZIONE 2012

Prefazione di Enrico Gregorio

Gruppo Utilizzatori Italiani di T_EX e L^AT_EX

Lorenzo Pantieri · Tommaso Gordini

L'arte di scrivere con L^AT_EX

Copyright © 2008-2012

COLOPHON

Questo lavoro è stato realizzato con L^AT_EX su Mac usando ArsClassica, una rielaborazione dello stile ClassicThesis di André Miede ispirato a *Gli elementi dello stile tipografico* di Robert Bringhurst.

I nomi commerciali, i loghi e i marchi registrati menzionati nella guida appartengono ai rispettivi proprietari, i pacchetti e le relative documentazioni ai rispettivi autori.

La copertina riproduce l'incisione *Tassellazione del piano con uccelli* di Maurits Cornelis Escher.

Il frontespizio riproduce la medesima incisione presente sulla copertina e la litografia *Mano con sfera riflettente*, dello stesso autore (tutte le riproduzioni delle opere di Escher contenute nella guida sono tratte da  **ESCHER**).

CONTATTI

 **GUIT** · Sito ufficiale del $\text{C}_U\text{I}\text{T}$

 lorenzo.pantieri@iperbole.bologna.it · Scrivi a Lorenzo Pantieri

 illinguista1972@gmail.com · Scrivi a Tommaso Gordini

La citazione è un utile sostituto dell'arguzia.

– Oscar Wilde

Dedicato a tutti gli appassionati di \LaTeX .

INDICE

PREFAZIONE XIII

INTRODUZIONE XVII

1	STORIA E FILOSOFIA	1
1.1	Storia	1
1.1.1	TeX	1
1.1.2	Etimologia	2
1.1.3	L ^A T _E X	2
1.2	Filosofia	3
1.2.1	Composizione sincrona e asincrona	3
1.2.2	Istruzioni di marcatura	4
1.2.3	L ^A T _E X: pro e contra	5
1.2.4	Luoghi comuni	7
1.2.5	Non tutte le strade portano a L ^A T _E X	8
1.2.6	L ^A T _E X: errori da evitare	8
2	INSTALLARE E AGGIORNARE	11
2.1	Ferri del mestiere	11
2.1.1	Editor e visualizzatore: TeXworks	11
2.1.2	TeX Live	14
2.2	Windows	15
2.2.1	Installare	15
2.2.2	Aggiornare	15
2.3	Mac OS X	16
2.3.1	Installare	16
2.3.2	Aggiornare	16
2.4	Linux	16
3	BASI	17
3.1	Per cominciare	17
3.2	Codifiche e lingue	19
3.2.1	Le codifiche di L ^A T _E X	19
3.2.2	Problemi con le codifiche	21
3.2.3	L ^A T _E X e le lingue	21
3.3	File con cui si ha a che fare	22
3.4	File sorgente	22
3.4.1	Comandi e ambienti	22
3.4.2	Caratteri speciali	25
3.4.3	Struttura del file sorgente	26
3.4.4	Spazi e righe vuote	27
3.4.5	Commenti	28
3.4.6	Sorgenti ordinati	28
3.5	Classi di documento	29
3.6	Gestire la pagina	31
3.6.1	Il tormentone dei margini	31
3.6.2	Interlinea e riempimento della pagina	32
3.7	Strutturare il documento	33
3.7.1	Sezionamenti principali e numerazione	33
3.7.2	Altri sezionamenti	33
3.7.3	Materiale iniziale, principale e finale	33

3.7.4	Appendici	35
3.7.5	Struttura generale di un libro o una tesi	35
3.8	Stili di pagina	36
3.9	Indice generale, titoli e profondità	37
3.9.1	Indice generale, miniindici e indici abbreviati	37
3.9.2	Gestire i titoli	37
3.9.3	Regolare la profondità dell'indice generale	38
3.10	Riferimenti incrociati	39
3.11	Collegamenti ipertestuali e indirizzi elettronici	40
3.12	Pacchetti	41
3.12.1	Caratteristiche	41
3.12.2	Caricamento e precauzioni	42
3.12.3	Usarli al meglio: la documentazione	43
3.12.4	E i file .ins e .dtx?	43
3.13	Unità di misura tipografiche	43
3.14	Documenti di grandi dimensioni	44
3.14.1	Il metodo di <code>\input</code>	45
3.14.2	Il metodo di <code>\include</code> e <code>\includeonly</code>	46
4	TESTO	47
4.1	Struttura del testo	47
4.2	Comporre i capoversi	47
4.3	Caratteri particolari e simboli	52
4.3.1	Virgolette, tratti e puntini di sospensione	52
4.3.2	Loghi, caratteri particolari, apici e pedici	53
4.4	Modificare stile e corpo del font	54
4.4.1	Modificare lo stile	54
4.4.2	Modificare il corpo	55
4.5	Titoli e frontespizi	56
4.6	Note a margine e a piè di pagina	57
4.7	Evidenziare le parole	58
4.8	Ambienti testuali	59
4.8.1	Elenchi puntati, numerati e descrizioni	59
4.8.2	Allineare e centrare i capoversi	60
4.8.3	Citazioni	61
4.8.4	Poesie	62
4.8.5	Codici e algoritmi	63
4.9	Acronimi e glossari	64
5	MATEMATICA	65
5.1	Formule in linea e in display	65
5.1.1	Formule in linea	65
5.1.2	Formule in display	66
5.1.3	Modo matematico e modo testuale	67
5.2	Nozioni introduttive	67
5.2.1	Raggruppamenti	68
5.2.2	Esponenti, indici e radici	68
5.2.3	Somme, prodotti e frazioni	68
5.2.4	Limiti, derivate e integrali	69
5.2.5	Insiemi numerici	70
5.2.6	Lettere greche	70
5.2.7	Simboli che sormontano altri simboli	71
5.2.8	Barre e accenti	71
5.2.9	Punti, frecce e simboli logici	73
5.2.10	Spazi in modo matematico	74

5.3	Operatori	75
5.3.1	Caratteristiche generali	75
5.3.2	Definire nuovi operatori	76
5.4	Parentesi	77
5.5	Vettori e matrici	79
5.6	Spezzare formule lunghe	80
5.6.1	Senza incolonnamento: <code>multline</code>	81
5.6.2	Con incolonnamento: <code>split</code>	81
5.7	Raggruppare più formule	82
5.7.1	Senza incolonnamento: <code>gather</code>	82
5.7.2	Con incolonnamento: <code>align</code>	82
5.7.3	Gli ambienti <code>gathered</code> e <code>aligned</code>	82
5.7.4	Casi e sottonumerazioni	83
5.8	Modificare stile e corpo del font	83
5.9	Evidenziare formule	85
5.10	Enunciati e dimostrazioni	86
5.10.1	Enunciati	86
5.10.2	Dimostrazioni	88
5.11	Diagrammi commutativi e di Venn	89
5.12	Altre scienze	89
6	TABELLE E FIGURE	91
6.1	Strumenti fondamentali	91
6.2	Oggetti in testo e fuori testo	92
6.2.1	Tabelle e figure in testo	92
6.2.2	Tabelle e figure fuori testo	93
6.3	Tabelle	97
6.3.1	Indicazioni generali	97
6.3.2	Tabelle standard: <code>tabular</code> e <code>array</code>	99
6.3.3	Celle con testo troppo lungo	100
6.3.4	Colonne di soli numeri: <code>siunitx</code>	102
6.3.5	Celle multicolonna	104
6.3.6	Celle multiriga	104
6.3.7	Spaziare a mano righe e colonne	105
6.3.8	Personalizzare le colonne: <code>array</code>	106
6.3.9	Tabelle con note	109
6.3.10	Tabelle grandi	109
6.4	Figure	113
6.4.1	Immagini vettoriali e bitmap	113
6.4.2	Convertire i formati	113
6.4.3	Ritagliare le immagini	114
6.4.4	Alcuni programmi utili	114
6.4.5	Includere le immagini nel documento	115
6.5	Disposizioni particolari	116
6.5.1	Didascalie laterali: <code>sidecap</code>	116
6.5.2	Oggetti multipli: <code>subfig</code>	117
6.5.3	Oggetti immersi nel testo: <code>wrapfig</code>	119
7	BIBLIOGRAFIA	121
7.1	Bibliografia manuale	121
7.2	Bibliografia automatica	123
7.2.1	Biber	124
7.2.2	Database bibliografici	124
7.2.3	Record e campi	125
7.2.4	Stili bibliografici e schemi di citazione	128
7.2.5	Comandi per le citazioni	131

7.2.6	Generare e collocare la bibliografia	132
7.2.7	Specialità	133
8	INDICE ANALITICO	137
9	CODICI	139
9.1	Impostazioni	139
9.2	Codici in linea e in display	139
9.2.1	Codici in linea	140
9.2.2	Codici in display	140
9.3	Codici mobili	141
9.4	Gestire i diversi linguaggi	142
9.5	Personalizzare l'aspetto dei codici	143
9.5.1	Numerare le righe	144
9.5.2	Rientri	145
9.5.3	Riquadri	146
9.5.4	Sfondi colorati	146
9.5.5	Evidenziare parole	147
9.5.6	Indicizzare parole di un codice	147
9.5.7	Allineare le colonne	148
9.6	Tecniche avanzate	148
10	GRAFICI	149
10.1	Grafici e tipografia	149
10.2	Il pacchetto pgfplots	149
10.2.1	Impostare il sistema di riferimento: l'ambiente axis	150
10.2.2	Disegnare il grafico: i comandi <code>\addplot</code> e <code>\addplot3</code>	154
10.3	Funzioni espresse analiticamente	155
10.3.1	Funzioni reali d'una variabile reale	155
10.3.2	Curve in forma parametrica	160
10.3.3	Funzioni reali di due variabili reali	162
10.3.4	Superfici in forma parametrica	165
10.4	Curve e superfici date per coordinate	166
10.5	Curve e superfici campionate da file	168
10.6	Altri sistemi di riferimento	169
10.7	Diagrammi a barre	171
10.7.1	Ortogrammi	171
10.7.2	Istogrammi	172
11	PERSONALIZZAZIONI	173
11.1	Comandi e ambienti personali	173
11.1.1	Nuovi comandi	173
11.1.2	Nuovi ambienti	174
11.2	Personalizzare il testo	175
11.2.1	Font	175
11.2.2	Lingue straniere	177
11.3	Specialità	177
11.3.1	Epigrafi	177
11.3.2	Capilettera	178
11.3.3	Scritture curiose	178
11.3.4	Documenti multicolonna	179
11.3.5	Colori	179
11.3.6	Filigrane	180
11.4	Creare nuovi oggetti mobili	180
11.5	Altre personalizzazioni	182
11.5.1	Modificare le parole fisse	182

11.5.2	Aggiungere spazio tra le voci dell'indice generale	183
11.5.3	Personalizzazioni avanzate	183
12	REVISIONE FINALE	185
12.1	Problemi orizzontali	185
12.2	Problemi verticali	186
A	NORME TIPOGRAFICHE	191
A.1	Accento e apostrofo	191
A.2	Punteggiatura e spaziatura	192
A.2.1	Segni d'interpunzione	192
A.2.2	Virgolette	192
A.2.3	Parentesi	193
A.2.4	Puntini di sospensione	193
A.2.5	Trattino, tratto e lineetta	193
A.2.6	Barra e asterisco	194
A.3	Stile del font	194
A.3.1	Corsivo	194
A.3.2	Nero	195
A.3.3	Maiuscoletto	195
A.4	Trattamento del testo	195
A.4.1	Parole straniere	195
A.4.2	Numeri	195
A.4.3	Frazioni, percentuali, unità di misura	197
A.4.4	Acronimi e abbreviazioni	197
B	LINGUE ANTICHE	199
B.1	Scrivere in latino	199
B.1.1	Marche prosodiche	199
B.1.2	Latino medievale	200
B.1.3	Documenti interamente in latino	200
B.2	Scrivere in greco	201
B.2.1	Traslitterare i caratteri	202
B.2.2	Scrivere in greco nel sorgente	204
B.2.3	Documenti interamente in greco	206
B.2.4	Specialità: teubner	206
C	VIDEOPRESENTAZIONI	207
C.1	Introduzione	207
C.2	Fondamentali	207
C.2.1	Per cominciare	207
C.2.2	Una presentazione articolata	208
C.2.3	Esposizione incrementale	212
C.2.4	Blocchi	214
C.2.5	Bibliografia	214
C.2.6	Codici	215
C.2.7	Disporre il contenuto su più colonne	215
C.2.8	Stampare la presentazione	216
C.3	Presentazioni personalizzate	217
	ACRONIMI	219
	ELENCO DEI SITI INTERNET	221
	BIBLIOGRAFIA	223
	INDICE ANALITICO	225

ELENCO DELLE FIGURE

Figura 1	Donald Knuth	1	
Figura 2	Vecchia dispensa di matematica	2	
Figura 3	Leslie Lamport	3	
Figura 4	Schermata di T _E Xworks	14	
Figura 5	Breve documento	26	
Figura 6	Esempio d'uso di frontespizio	57	
Figura 7	Figura collocata a mano	97	
Figura 8	Esempio d'uso di sidecap	117	
Figura 9	Esempio d'uso di subfig	118	
Figura 10	Esempio d'uso di wrapfig	119	
Figura 11	Alcuni stili bibliografici	130	
Figura 12	Bibliografie suddivise	134	
Figura 13	Sistemi di riferimento predefiniti di pgfplots	152	
Figura 14	Definizione di azimuth ed elevazione	164	
Figura 15	Famiglie di font a larghezza variabile	176	
Figura 16	Famiglie di font a larghezza fissa	176	
Figura 17	Tastiera italo-greca	202	
Figura 18	Presentazione semplice	209	
Figura 19	Presentazione complessa	211	
Figura 20	Esempi di blocchi	213	
Figura 21	Diapositive particolari	214	
Figura 22	Diapositive su più colonne	216	
Figura 23	Alcuni temi predefiniti di beamer	217	

ELENCO DELLE TABELLE

Tabella 1	Cronologia di T _E X e L ^A T _E X	3	
Tabella 2	Percorsi di T _E X	12	
Tabella 3	Modi di interazione tra L ^A T _E X e utente	18	
Tabella 4	File ausiliari	22	
Tabella 5	Principali tipi di comando	23	
Tabella 6	Caratteri speciali di L ^A T _E X	24	
Tabella 7	Scorciatoie da tastiera	25	
Tabella 8	Opzioni delle classi standard	31	
Tabella 9	Comandi di sezionamento	34	
Tabella 10	Struttura generale di un libro o una tesi	35	
Tabella 11	Elementi richiamabili in un documento	38	
Tabella 12	Corrispondenza fra livelli e sezioni	39	
Tabella 13	Unità di misura tipografiche	44	
Tabella 14	Lunghezza orientativa delle sezioni	48	
Tabella 15	Virgolette, tratti e puntini di sospensione	52	
Tabella 16	Loghi frequenti	53	
Tabella 17	Accenti e caratteri particolari	54	
Tabella 18	Comandi per modificare lo stile del font	55	
Tabella 19	Dichiarazioni per modificare il corpo del font	56	
Tabella 20	Simboli insiemistici	70	

Tabella 21	Lettere greche	71	
Tabella 22	Accenti in modo matematico		72
Tabella 23	Frecce	73	
Tabella 24	Simboli logici	74	
Tabella 25	Spazi in modo matematico		75
Tabella 26	Operatori predefiniti	77	
Tabella 27	Simboli speciali	80	
Tabella 28	Simboli di relazione		81
Tabella 29	Stili matematici	84	
Tabella 30	Preferenze di collocazione per gli oggetti mobili		96
Tabella 31	Tabella che non rispetta le regole generali		97
Tabella 32	Tabella che rispetta le regole generali		97
Tabella 33	Descrittori standard per le tabelle		99
Tabella 34	Tabella con colonna p	100	
Tabella 35	Tabella tabularx	101	
Tabella 36	Tabella tabularx con più colonne X		101
Tabella 37	Tabella con colonna S	102	
Tabella 38	Tabella con cella multicolonna		103
Tabella 39	Tabella con celle multiriga	104	
Tabella 40	Tabella con celle multiriga e multicolonna		105
Tabella 41	Tabelle con colonne p diversamente allineate		108
Tabella 42	Tabella array	109	
Tabella 43	Tabella con corpo di carattere ridotto		110
Tabella 44	Esempio di tabella ripartita su più pagine		111
Tabella 45	Alcuni programmi utili per lavorare con L ^A T _E X		114
Tabella 46	Principali chiavi di graphicx	116	
Tabella 47	Preferenze di collocazione di wrapfloat		120
Tabella 48	Sintassi di <code>\index</code>	138	
Tabella 49	Alcuni linguaggi riconosciuti da listings		143
Tabella 50	Valori di frame e loro comportamento		146
Tabella 51	Sistemi di riferimento disponibili in pgfplots		150
Tabella 52	Funzioni e costanti matematiche con pgfplots		154
Tabella 53	Spessori di linea disponibili in pgfplots		156
Tabella 54	Colori disponibili in pgfplots	157	
Tabella 55	Tratti disponibili in pgfplots	158	
Tabella 56	Alcuni marcatori disponibili in pgfplots		167
Tabella 57	Parole fisse italiane di babel	182	
Tabella 58	Diacritici e punteggiatura del greco antico		203

PREFAZIONE

La storia di $\text{T}_{\text{E}}\text{X}$ in Italia è lunga: secondo i resoconti del $\text{T}_{\text{E}}\text{X}$ Users Group, il primo convegno su $\text{T}_{\text{E}}\text{X}$ tenuto in Europa è stato a Como nel maggio 1985 e ne esiste perfino un volume di rendiconti a cura di Dario Lucarella.

Per lunghi anni, però, è mancato un manuale in italiano che spiegasse ai possibili utenti di $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ come cominciare a scrivere documenti con questo sistema. Da pochi anni è disponibile la traduzione italiana della *Not So Short Introduction to $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$* , che però è ferma a una versione abbastanza antiquata. Si trovano nei meandri di Internet altre guide, alcune scritte fin dall'inizio in italiano, altre tradotte. Tutte, compresa la *Not So Short*, soffrono di un grave difetto: rispecchiano in gran parte il modo in cui il loro autore ha imparato $\text{T}_{\text{E}}\text{X}$ e $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

Ogni utente di $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ha sviluppato le proprie tecniche, raccolte qua e là in modo spesso disordinato, e difficilmente si adatta a cambiare anche se ne scopre di nuove e anche più efficienti. Chiunque abbia provato a raccogliere documenti scritti da diverse persone se ne rende conto a una prima occhiata: molti di questi documenti hanno trovato posto nelle raccolte di *errori*. Sia chiaro, nessuno è immune dagli *errori*, compreso chi scrive: quando penso a come scrivevo codice $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ anni fa mi vengono i brividi. Quando leggo una guida a $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ la prima cosa che faccio è cercare gli *errori* e inevitabilmente li trovo, così come li si trova molto facilmente nei preamboli che circolano fra chi scrive tesi di laurea o di dottorato.

Non vorrei dare l'impressione di criticare gente che ha speso molto tempo per mettere a disposizione di tutti le proprie conoscenze: ogni guida ha pregi e difetti, solo che i difetti hanno la curiosa tendenza di diffondersi più dei pregi. Anche nel mondo $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ evidentemente vale il principio che la moneta cattiva scaccia quella buona.

Che dire di questa guida scritta da Lorenzo Pantieri? Che forse è l'unica che affronta alla radice il problema descritto. Ciò che a prima impressione può apparire pignoleria è invece profonda curiosità di andare alla ricerca del modo migliore per risolvere i problemi. Mi riferisco alle innumerevoli domande poste da Lorenzo sul forum del $\text{G}_{\text{U}}\text{I}$ su questioni stilistiche, su come affrontare problemi tipografici o più strettamente riguardanti $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Questa curiosità non è pedanteria fine a sé stessa: è desiderio di esplorare il linguaggio $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ per trovare sempre la soluzione più efficace.

Il risultato è una guida molto piacevole da leggere, che introduce al linguaggio $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ in modo chiaro evitando per quanto possibile le complicazioni che spesso si trovano in altri scritti simili. Certo, non è completa: se pensiamo che il *$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ Companion* ha più di mille pagine, siamo molto distanti da questo obiettivo, che lo stesso autore dice di non avere. Nessuna guida potrà mai essere completa, visto che $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ è usato per comporre documenti di generi diversissimi, dalla matematica alla fisica, al diritto o alle edizioni critiche di testi classici. In una guida introduttiva, soprattutto a $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, è importante la chiarezza nella spiegazione dei motivi che impongono all'utente certe scelte, con l'indicazione di dove trovare le soluzioni ai problemi che nella guida stessa sono solo accennati.

Devo menzionare l'eccellente guida scritta con grande competenza da Claudio Beccari, che però è rivolta a un pubblico diverso: sicuramente chi avrà letto *L'arte di scrivere con $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$* sarà in grado, se lo desidera, di affrontare gli argomenti più complessi oggetto di quell'opera.

Questa è solo la prima edizione: possiamo essere sicuri che Lorenzo sarà sempre pronto ad aggiunte e correzioni in modo che chi la legge trovi facilmente la risposta ai propri problemi. Sarà un piacere provare a dare una mano sul forum in modo che questa guida diventi ancora migliore e conquistisi sempre nuovi lettori contribuendo alla diffusione di \TeX e \LaTeX in Italia.

Padova, 30 marzo 2008

Enrico Gregorio

SU QUESTA NUOVA EDIZIONE

La seconda edizione dell'*Arte* ha la grande novità di un secondo autore, Tommaso Gordini, la cui pignoleria linguistica è pari alla pignoleria \TeX nica di Lorenzo Pantieri. Il risultato è una profonda revisione del testo: anche grazie ai consigli e suggerimenti dei numerosi lettori, questa guida è arrivata a una maturità che la rende indispensabile a chi voglia imparare a maneggiare \LaTeX con facilità. Gli autori perdoneranno la parola "pignoleria" usata con grande affetto: è caparbia e ostinazione nella ricerca del modo migliore di trasmettere i concetti al lettore, evitando come la peste i tecnicismi e le parole arcane tanto frequenti nei manuali di software. Credo che l'obiettivo sia stato raggiunto con successo.

Resta solo da augurare agli autori un buon riposo in vista della terza edizione. Il mondo \TeX è sempre in fermento: $X_{\text{Y}}\TeX$ e $\text{Lua}\TeX$ promettono nuovi traguardi e possibilità per chi ama comporre i propri testi secondo i canoni dell'arte tipografica tramandata da secoli.

Padova, 16 luglio 2010

Enrico Gregorio

SULL'EDIZIONE 2012

La guida è stata aggiornata nelle parti consuete e ulteriormente rivista. Pochi, quest'anno, gli argomenti espunti. Piuttosto importanti e consistenti, invece, le aggiunte. L'impianto complessivo dell'opera, come al solito, rimane invariato.

Non è ancora la terza edizione preconizzata due anni addietro dal nostro Egregio prefatore, questa, ma facciamo nostro anche quest'anno l'augurio di buon riposo. A tutti, ormai si sa, il nostro più sonoro «buona lettura!».

Cesena-Padova, 22 novembre 2012

Gli Autori

RINGRAZIAMENTI

Desidero ringraziare innanzitutto i membri dello staff del $\mathcal{G}\mathcal{I}\mathcal{T}$ e poi tutti quelli che hanno discusso con me sul forum del Gruppo, in particolare Fabiano Busdraghi, Gustavo Cevolani, Rosaria D'Addazio, Massimiliano Dominici, Gloria Faccanoni, Gianluca Gorni, Maurizio Himmelmann, Jerónimo Leal, Lapo Filippo Mori, Andrea Tonelli, Emiliano Giovanni Vavassori ed Emanuele Vicentini, per l'insostituibile aiuto fornito nella redazione di questo lavoro, le spiegazioni dettagliate, la pazienza e la precisione nei suggerimenti, le soluzioni fornite, la competenza e la disponibilità: grazie mille, ragazzi!

Rivolgo un ringraziamento davvero particolare al professor Enrico Gregorio, per i suoi impagabili insegnamenti e per avermi concesso l'onore di scrivere la prefazione alla guida.

Un "grazie" altrettanto speciale va al professor Claudio Beccari, per i consigli durante la revisione di un'opera che senza la sua pazienza non avrebbe mai raggiunto la forma attuale.

Ringrazio infine André Miede, per aver realizzato lo stile ClassicThesis (con cui è composto questo lavoro), Daniel Gottschlag, che mi ha dato lo spunto per realizzarne un'originale rielaborazione, e Ivan Valbusa, per i consigli grafici, le idee e per aver creato lo stile con cui è composta la bibliografia di questa guida.

Lorenzo Pantieri

Un libro come questo non può *mai* dirsi veramente concluso, e per fortuna il vuoto lasciato dalla fine del lavoro non è così grande. Ai ringraziamenti di Lorenzo, che faccio miei, dovrei aggiungere quelli a un numero imprecisato di persone: lo faccio silenziosamente, e so che non me ne vorranno per non vedere il proprio nome in queste righe.

Voglio però nominare e ringraziare, non necessariamente in quest'ordine, le quattro impigliatesi più saldamente nella rete.

I miei genitori: devo a loro il fatto di capirci (qualche volta) qualcosa. Direi che hanno fatto un buon lavoro.

Alberto, che per primo mi disse: «infatti non è Word, è *lātek!*». Non nascondo che lì per lì l'osservazione mi lasciò del tutto indifferente.

Enrico, il saggio Vecchio Lupo che ha una pazienza grande come il sole e che (ancora) non perde il pelo. Il resto lo conosce.

Eleonora, adorabile e impotente compagna in quest'avventura: senza di lei non potreste leggere quello che state leggendo. Questo libro è più suo che mio.

Tommaso Gordini

Cesena-Padova, 16 luglio 2010

INTRODUZIONE

*Abbiamo visto che la programmazione è un'arte,
perché richiede conoscenza, applicazione, abilità e ingegno,
ma soprattutto per la bellezza degli oggetti che produce.*

– Donald Ervin Knuth

\LaTeX è un programma di composizione tipografica liberamente disponibile, indicato soprattutto per scrivere documenti scientifici con la più alta qualità. Lo scopo di questo lavoro, rivolto sia a chi muove i primi passi in \LaTeX sia a quanti già lo conoscono, è di offrire ai suoi utenti italiani le conoscenze essenziali per poterlo usare con successo.

I concetti fondamentali della materia, raccolti da svariati manuali, vengono presentati nel modo più chiaro e organico possibile; nel contempo si fornisce un vasto campionario di esempi e si analizzano alcuni tipici problemi che potrebbero presentarsi nella redazione di una pubblicazione scientifica o professionale in italiano, indicando per ciascuno le soluzioni per noi migliori.

Queste derivano principalmente dalle numerose discussioni presenti sul forum del GUIT (Gruppo Utilizzatori Italiani di $\text{T}_{\text{E}}\text{X}$ e \LaTeX , GUIT), che resta sempre un eccellente riferimento per tutti i temi trattati in questa guida, la cui lettura, lo si ricorda, presuppone la conoscenza almeno basilare del calcolatore e di Internet.

In linea generale si è seguita la prassi di non scandagliare troppo i vari argomenti: dei pacchetti citati (ma non di tutti), infatti, si analizzano soltanto le impostazioni più importanti e se ne suggerisce l'uso, indirizzando alla relativa documentazione chi voglia approfondirne la conoscenza.

L'esposizione degli argomenti è articolata come segue.

IL PRIMO CAPITOLO traccia una breve storia di \LaTeX , indicandone idee di fondo e peculiarità.

IL SECONDO CAPITOLO spiega come installare e aggiornare \LaTeX sul proprio calcolatore.

IL TERZO CAPITOLO presenta alcune nozioni fondamentali indispensabili per comprendere il funzionamento del programma: la sua lettura, perciò, è propedeutica a quella del resto della guida.

IL QUARTO CAPITOLO descrive gli strumenti per trattare il testo.

IL QUINTO CAPITOLO esplora uno dei principali punti di forza di \LaTeX : la composizione di formule matematiche.

IL SESTO CAPITOLO presenta i concetti e gli strumenti essenziali per comporre le tabelle, includere le figure in un documento e gestire la collocazione degli oggetti mobili sulla pagina.

IL SETTIMO CAPITOLO presenta gli strumenti per realizzare e gestire la bibliografia.

L'OTTAVO CAPITOLO illustra le nozioni essenziali per generare l'indice analitico.

IL NONO CAPITOLO riguarda la scrittura di codici e linguaggi di programmazione in un documento \LaTeX .

IL DECIMO CAPITOLO si occupa della rappresentazione grafica di dati.

L'UNDICESIMO CAPITOLO espone alcuni suggerimenti per fare in modo che \LaTeX produca risultati diversi da quelli predefiniti.

IL DODICESIMO CAPITOLO dà alcuni suggerimenti per migliorare l'impaginazione del documento.

L'APPENDICE A descrive sinteticamente le principali norme tipografiche italiane.

L'APPENDICE B presenta gli strumenti per scrivere in latino e in greco.

L'APPENDICE C, infine, fornisce gli elementi essenziali per realizzare una videopresentazione.

Questo *non* è un manuale su \LaTeX , ma piuttosto un tentativo di riordinare in forma scritta appunti accumulatisi nel tempo, via via che divenivamo abituali utenti di questo programma. In qualità di semplici appassionati non abbiamo nulla da insegnare; d'altra parte abbiamo studiato \LaTeX e l'abbiamo usato intensamente, acquisendo una certa esperienza che ci piacerebbe condividere con altri utenti.

Se avete idee su argomenti da aggiungere, togliere o modificare in questo libro, o se vi dovesse capitare di notare errori, sia di battitura che di sostanza (ed è probabile che ce ne siano parecchi, e del primo e del secondo tipo), fareste davvero una cosa gradita comunicandocelo, così da poterli correggere nelle versioni successive del lavoro. In modo particolare, ci interessano i commenti di chi è alle prime armi su quali parti di questa guida sono di più facile comprensione e quali invece potrebbero essere spiegate meglio.

Ecco lo spirito che ci ha guidati in questo lavoro: speriamo che possiate usare \LaTeX con il nostro stesso piacere.

1

STORIA E FILOSOFIA

Questo capitolo presenta una breve storia di \TeX e \LaTeX , e ne indica idee di fondo e peculiarità. Se ne descrivono inoltre i principali pro e contra, i luoghi comuni, le scorciatoie da non prendere e i più comuni errori da evitare all'inizio.

1.1 STORIA

1.1.1 \TeX

\TeX è un programma di composizione tipografica realizzato da Donald Ervin Knuth, professore di Informatica all'Università di Stanford (USA), e distribuito con una licenza di software libero.

Nel 1977, Knuth cominciò a scrivere il suo "motore" di tipocomposizione per esplorare le potenzialità degli strumenti di stampa digitale che allora cominciavano a muovere i primi timidi passi in campo editoriale.

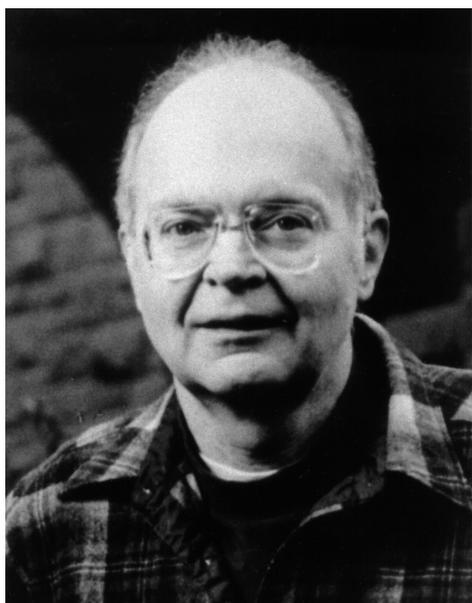


Figura 1: Donald Ervin Knuth (l'immagine è tratta da  [KNUTH](#)).

La qualità tipografica delle proprie pubblicazioni (e soprattutto del capolavoro *The Art of Computer Programming*, in più volumi ricchi di formule matematiche), constatava Knuth, era in crescente peggioramento: a quel tempo, infatti, gran parte della matematica si componeva con la macchina per scrivere, alzando e abbassando il carrello per indici ed esponenti e cambiando testina per i simboli (si veda la figura 2 nella pagina seguente). Con \TeX , il Professore sperava di mettere un freno a questa tendenza.

Il programma ha visto la luce nel 1982, e da allora ha subito costanti aggiornamenti e miglioramenti tra cui quello del 1989, che gli ha permesso di supportare un numero considerevolmente maggiore di caratteri (anche non

latini). \TeX è rinomato per essere estremamente stabile e per girare su diversi tipi di calcolatore, ed è stato rivisto per l'ultima volta nel 2008. Il suo numero di versione converge a π (attualmente è 3,1415926).

The \TeX book [Knuth, 1984], scritto da Donald Knuth, è il manuale d'uso di \TeX e uno dei libri più completi su questo programma. Attualmente \TeX è un marchio registrato della \mathcal{AMS} (American Mathematical Society).



Figura 2: Pagina di una dispensa di matematica degli anni Settanta del secolo scorso, dattilografata, con aggiunte a mano.

1.1.2 Etimologia

Knuth ha nascosto un trabocchetto nel nome del programma: $\text{T}_{\text{E}}\text{X}$, infatti, si pronuncia *tèch* (aspirando il *ch* finale) e non com'è scritto, perché è una parola greca scritta in greco maiuscolo (in lettere minuscole si scriverebbe $\tau\epsilon\chi$). Si tratta di un'antichissima radice indoeuropea comune non soltanto ai greci $\tau\epsilon\chi\tau\omega\nu$ (pron. *tèkton*, "artefice") e $\tau\epsilon\chi\nu\eta$ (pron. *tèchne*, insieme "arte" e "mestiere"), per esempio, ma viva ancora oggi negli usatissimi *tecnica*, *politecnico*, *architetto* e in numerose altre parole italiane. L'etimologia appena spiegata illumina la scelta di Knuth: $\text{T}_{\text{E}}\text{X}$ sarebbe stato il nome perfetto per un programma che compone documenti "allo stato dell'arte".

Knuth dice che «se $\text{T}_{\text{E}}\text{X}$ è ben pronunciato, lo schermo del calcolatore si appanna leggermente». La lettera *X*, infatti, corrisponde a una forte aspirazione non esistente in italiano, ma presente in molte lingue attualmente parlate nel mondo: oltre che in greco, si trova nel tedesco *Bach*, nello scozzese *Loch*, nello spagnolo *Juan* e *Mexico*, nel russo *Хорошо* (pron. *harasciò*, "bene"), nel cinese 你好 (pron. *nǐ hǎo*, "ciao"), solo per fare qualche esempio.

Lo stesso Knuth, però, ammette le diverse pronunce nazionali (che convergono in un *tèk* universale), rassicurando gli utenti che non è certo per il fatto di sentire $\text{T}_{\text{E}}\text{X}$ detto come ognuno preferisce che andrà su tutte le furie. Tuttavia, indica nella Grecia il Paese in cui oggi si può ascoltare la pronuncia più corretta di questa parola.

1.1.3 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ($\text{L}^{\text{A}}(\text{mport})\text{T}_{\text{E}}\text{X}$) è un programma di composizione tipografica realizzato da Leslie Lamport e liberamente disponibile, che usa $\text{T}_{\text{E}}\text{X}$ come motore di tipocomposizione.

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ non è $\text{T}_{\text{E}}\text{X}$, però. Per dare l'idea della differenza tra i due programmi, si potrebbe paragonare $\text{T}_{\text{E}}\text{X}$ a un corpo, e $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ al più popolare degli "abiti" (fatto, però, di istruzioni in linguaggio $\text{T}_{\text{E}}\text{X}$) che nel corso degli anni gli

Tabella 1: Cronologia di $\text{T}_\text{E}\text{X}$ e $\text{L}\text{A}\text{T}_\text{E}\text{X}$.

1977	Knuth comincia a scrivere $\text{T}_\text{E}\text{X}$
1978	Lamport comincia a scrivere il primo nucleo di $\text{L}\text{A}\text{T}_\text{E}\text{X}$
1982	Prima versione pubblica di $\text{T}_\text{E}\text{X}$
1984	Knuth pubblica la prima edizione di <i>The $\text{T}_\text{E}\text{X}$book</i>
1985	Lamport pubblica $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 2.09 e la prima edizione di <i>$\text{L}\text{A}\text{T}_\text{E}\text{X}$. A Document Preparation System</i>
1994	$\text{L}\text{A}\text{T}_\text{E}\text{X}$ 2 ϵ
2008	Ultima revisione di $\text{T}_\text{E}\text{X}$
2011	Ultima revisione di $\text{L}\text{A}\text{T}_\text{E}\text{X}$

sono stati confezionati addosso per avvicinarlo al pubblico con sembianze amichevoli.

$\text{L}\text{A}\text{T}_\text{E}\text{X}$ è stato progettato per automatizzare in una volta sola tutte le più comuni operazioni necessarie per realizzare un documento e, tramite le proprie impostazioni predefinite, permette all'utente di impaginare il suo lavoro *ai più elevati livelli di professionalità secondo canoni tipografici consolidati*.

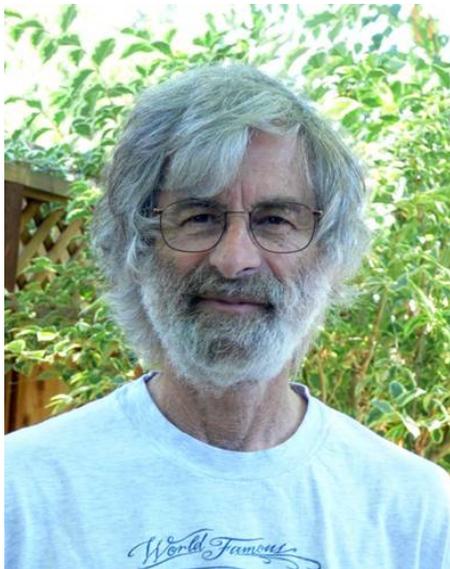


Figura 3: Leslie Lamport (l'immagine è tratta da \LaTeX LAMPORT).

Lamport, che collaborava con Knuth allo sviluppo di $\text{T}_\text{E}\text{X}$, cominciò a scrivere $\text{L}\text{A}\text{T}_\text{E}\text{X}$ alla fine degli anni Settanta del secolo scorso, quando $\text{T}_\text{E}\text{X}$ non era ancora stato pubblicato.

La prima versione pubblica di $\text{L}\text{A}\text{T}_\text{E}\text{X}$ risale al 1985, e da allora il programma è stato continuamente aggiornato e migliorato. Per molti anni il suo numero è rimasto fissato a 2.09 e le successive revisioni sono state identificate con le loro date.

Nel 1994, finalmente, un gruppo di programmatori guidato da Frank Mittelbach lo ha aggiornato in modo sostanziale, includendovi tutte le versioni successive alla 2.09 e numerosi altri miglioramenti. Per distinguerla da quella precedente, la nuova edizione è stata battezzata $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 2 ϵ , e costituisce l'oggetto di questa guida. *$\text{L}\text{A}\text{T}_\text{E}\text{X}$. A Document*

Preparation System [Lamport, 1994], scritto da Leslie Lamport, è il manuale ufficiale di $\text{L}\text{A}\text{T}_\text{E}\text{X}$.

Il futuro $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 3 si profila come un progetto a lungo termine: i costanti aggiornamenti di $\text{L}\text{A}\text{T}_\text{E}\text{X}$, la cui ultima versione pubblica è del 2011, ne costituiscono le tappe di avvicinamento.

1.2 FILOSOFIA

1.2.1 Composizione sincrona e asincrona

La caratteristica che più differenzia $\text{L}\text{A}\text{T}_\text{E}\text{X}$ dagli altri elaboratori di testo è il fatto che l'introduzione del testo e la sua composizione avvengono *in tempi diversi* [Beccari, 2012].

Per modificare un documento scritto con un comune *word processor* (come Microsoft Word), l'utente agisce direttamente sul testo già composto così come gli appare sul monitor, e ogni sua azione si traduce in una variazione *immediata* di quel testo. Perciò questo tipo di composizione è detto "composizione sincrona". Per essere *davvero* sincrono e operare con un ritardo trascurabile tra modifica del testo e visualizzazione, però, il programma deve puntare tutto sulla rapidità della presentazione, ciò che di fatto impedisce di ottenere una composizione perfetta per via della molto più accurata elaborazione del materiale richiesta. È anche vero, però, che oggi i programmi di videoscrittura sono estremamente rapidi e che la qualità della loro composizione migliora sensibilmente a ogni nuova versione, ma il compromesso tra velocità e qualità esiste sempre.

La "composizione asincrona", invece, consiste nell'introdurre il testo in un editor concentrandosi unicamente su struttura logica e contenuto del documento (senza perciò preoccuparsi di come apparirà), per darlo "in pasto" a un compositore (L^AT_EX, in questo caso) che lo impagina solo *successivamente*. L'utente, naturalmente, può modificare il proprio lavoro in qualunque modo anche dopo la composizione, ma sempre tenendo bene a mente che L^AT_EX non si limita ad aggiustarlo *nel punto* in cui è stato modificato e basta, come farebbe un programma qualunque, ma riorganizza sempre *l'intero capoverso* (e, di conseguenza, l'intera pagina) nel migliore dei modi.

Va da sé che questo secondo tipo di composizione è migliore del primo, perché la velocità di visualizzazione passa in secondo piano a tutto vantaggio della qualità: L^AT_EX elabora il testo sempre *nel suo complesso*, e per questo motivo può fare le scelte d'impaginazione migliori.

1.2.2 Istruzioni di marcatura

L'idea di L^Ampo^rt era forte. E L^AT_EX centra in pieno l'obiettivo: l'utente può (quasi *deve*, si direbbe) astrarsi dai dettagli estetici che con un altro elaboratore di testo sarebbe costretto a introdurre a mano, per indirizzare più efficacemente le proprie energie su *contenuto* e scansione delle parti del proprio lavoro.

L^AT_EX pretende dall'utente considerazioni sul *cosa*: «il mio documento sarà composto da un certo numero di capitoli, ciascuno diviso in paragrafi numerati, avrà indice generale e analitico, delle figure e qualche tabella». E questo è tutto.

Al *come* pensa L^AT_EX, e lo fa molto bene. Per esempio, uno stesso file sorgente può generare in teoria documenti radicalmente diversi soltanto cambiandone la classe o caricando un pacchetto che agisce in modo globale su di esso. (S'è detto *in teoria* perché, nella pratica, qualche aggiustamento manuale si rende di fatto *sempre* necessario.)

Un file da comporre con L^AT_EX è scritto in una "lingua" particolare costituita da *marcatori* (o *etichette logiche*, *mark-up* in inglese), ovvero istruzioni che il programma deve eseguire per trattare nel modo specificato dall'utente la porzione di testo a cui si riferiscono. Le etichette logiche sono, in una parola, i *comandi* (chiamati anche, in gergo, *macro*, abbreviazione di *macroistruzioni*) e gli *ambienti*.

Nonostante faccia dei marcatori uno dei propri punti di forza (per cui molte delle tradizionali operazioni sul testo vengono automatizzate), L^AT_EX definisce anche gli strumenti per regolare finemente (e a mano, questa volta) il risultato durante la revisione finale del documento, passaggio ancora insostituibile. In tipografia, infatti, le variabili sono troppe e troppo variegate per poter essere gestite in una volta sola da un unico programma; e l'utente deve accettarlo, se pretende un prodotto di alta qualità.

Il file prodotto con l'editor, insomma, è un codice scritto in una sorta di linguaggio di programmazione, dato che contiene sia il vero e proprio testo del documento, sia le istruzioni di marcatura (i comandi) che ordinano al programma di tipocomposizione di comporre quello che gli si dà in pasto secondo lo stile particolare scelto per scrivere il documento.

Non si spaventi chi non conosce bene l'inglese: i comandi di \LaTeX (che specificano l'inizio di un capitolo o una particolare modalità di comporre una formula matematica, per esempio), sono in un inglese molto semplice, e anche se abbreviati risultano di solito comprensibilissimi e facilmente memorizzabili.

Un esempio

Per dare l'idea di come appare un documento da comporre con \LaTeX , di seguito si riportano alcune righe di *codice sorgente* (o più semplicemente *sorgente*), cioè l'insieme di testo, numeri, simboli e istruzioni di marcatura da scrivere nell'editor.

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

Il sorgente viene composto da \LaTeX che, attraverso \TeX , produce il documento tipocomposto (*typeset*). Se il risultato non soddisfa, non si può modificare direttamente il documento a schermo, ma bisogna correggere il sorgente e poi ricomporlo.

L'esempio seguente riporta a sinistra lo stesso sorgente appena visto e a destra il risultato della composizione.

<pre>Due matrici $n \times n$ complesse A e B si dicono <i>simili</i> se esiste una matrice $n \times n$ invertibile T tale che \begin{equation} B=T^{-1}AT. \end{equation}</pre>	<p>Due matrici $n \times n$ complesse A e B si dicono <i>simili</i> se esiste una matrice $n \times n$ invertibile T tale che</p> $B = T^{-1}AT. \quad (1.1)$
--	--

Nei prossimi capitoli si spiegheranno tutte le istruzioni usate nell'esempio. Tuttavia, anche con pochi rudimenti di inglese si capisce facilmente quello che il linguaggio di marcatura ha specificato.

1.2.3 \LaTeX : pro e contra

I vantaggi di \LaTeX rispetto agli altri elaboratori di testo sono innumerevoli. Di seguito se ne elencano alcuni.

- \LaTeX compone documenti al massimo grado di professionalità e presenta caratteristiche di qualità e stabilità sconosciute agli altri elaboratori di testo.
- L'utente pensa a struttura e contenuto del documento, \LaTeX si occupa della sua impaginazione.
- \LaTeX genera strutture complesse come riferimenti incrociati, indici e bibliografie con grande efficienza e flessibilità.

- \LaTeX gestisce in modo impeccabile e tuttora ineguagliato la composizione tipografica di formule matematiche e capoversi.
- \LaTeX è gratuito, multilingue e multipiattaforma (il suo sorgente è puro testo, e dunque viene letto tranquillamente su tutti i sistemi operativi; si veda, però, il paragrafo 3.2.1 a pagina 19).
- \LaTeX ha una struttura modulare che permette di estenderne le capacità per eseguire compiti tipografici non direttamente gestiti dal programma.
- \LaTeX rende davvero difficile creare documenti mal strutturati e tipograficamente scadenti.
- \LaTeX appassiona talmente... che ci si può dimenticare la tazza del caffè sul foglio!

Innegabilmente, \LaTeX presenta anche alcuni svantaggi.

- Con \LaTeX ci vuole attitudine all'astrazione.
- Con \LaTeX la gratificazione non è istantanea, ma ritardata.
- \LaTeX «non lavora bene per chi ha venduto la propria anima» [Oetiker *et al.*, 2011].
- Solo gli esperti possono permettersi di uscire dagli stili predefiniti.
- Il fatto che \LaTeX raggiunga un'elevatissima qualità soltanto grazie alla composizione asincrona può talvolta rivelarsi un difetto anziché un pregio (almeno se non si considerano i motivi di questo comportamento e non si adatta di conseguenza il proprio metodo di lavoro). \LaTeX , infatti, gestisce il capoverso in modo del tutto particolare: rimuoverne anche una sola parola ne determina *sempre* la riorganizzazione completa. Con risultati a volte indesiderati: il capoverso potrebbe risultare addirittura di una riga più lungo, se \LaTeX decidesse che quella è la soluzione migliore, con le ovvie conseguenze se ciò dovesse accadere durante l'ultimissima revisione del documento.
- \LaTeX non gestisce pagine più grandi di 33 metri quadrati e non dà denaro alle case di software.

In breve, è opportuno usare \LaTeX quando [Mittelbach *et al.*, 2007]:

- l'utente preferisce pensare per strutture logiche;
- i documenti da preparare richiedono coerenza interna;
- i documenti da preparare non hanno un formato completamente definito o potrebbero essere presentati parallelamente in vesti diverse;
- i documenti da preparare contengono molta matematica;
- il materiale è corposo.

Viceversa, si sceglierà un sistema di videoscrittura tradizionale quando:

- l'utente preferisce pensare per strutture visive;
- l'utente non si trova del tutto a proprio agio nel lavorare con un linguaggio di programmazione (un editor per \LaTeX aiuta, ma...);

- i documenti da preparare richiedono più flessibilità visuale che consistenza (volantini, biglietti d'invito, dépliant, brochure, eccetera);
- il materiale non è corposo.

In ultima battuta: prima di affidarsi a \LaTeX si consiglia caldamente di vagliare *a monte* e con la massima attenzione il tipo di prodotto che s'intende realizzare.

1.2.4 Luoghi comuni

Molte delle persone che si avvicinano a \LaTeX dopo anni di tormenti alle prese con altri elaboratori di testo si stupiscono quando scoprono che era disponibile da oltre tre decenni, pur non avendone mai sentito parlare. Non si tratta certo di una cospirazione ai loro danni, ma solo di «un segreto ben conservato e noto solo a pochi milioni di persone», come disse una volta un utente anonimo.

Knuth e L^ampo^rt hanno generosamente reso i propri programmi di pubblico dominio fin da subito, e perciò per molto tempo \LaTeX è vissuto indisturbato tra le mura delle università senza che nessuno sentisse il bisogno di pubblicizzarlo al di fuori. Oggi, tuttavia, \LaTeX è divenuto popolarissimo: innumerevoli editori pubblicano documenti nel suo formato, e centinaia di migliaia di utenti scrivono con esso milioni di documenti ogni giorno.

In questi anni, su \LaTeX sono fioriti molti luoghi comuni: per evitare possibili incomprensioni, conviene in questa sede esaminarne i più diffusi.

Leggenda: \LaTeX ha solo un font

\LaTeX può usare *ogni* font di tipo TrueType, Type1, OpenType, PostScript e METAFONT. Ciò è più di quanto viene offerto dalla maggior parte degli altri sistemi di composizione tipografica. Il font standard di \LaTeX è il Computer Modern e non il Times New Roman, quindi alcuni restano turbati da un documento diverso dal solito.

Leggenda: \LaTeX è un software solo per Unix

Si sente anche dire che \LaTeX è un software solo per Unix, o solo per Mac, eccetera. È tutto il contrario: \LaTeX funziona sulla maggior parte dei calcolatori oggi sul mercato (supercomputer e palmari compresi): dai PC con Windows ai Mac ai sistemi Unix/Linux. Se \LaTeX non gira sul proprio computer è solo perché, molto probabilmente il sistema operativo installato è troppo vecchio.

Leggenda: \LaTeX è obsoleto

Proprio il contrario. L'incessante lavoro di migliaia di appassionati in tutto il mondo lo tiene costantemente aggiornato, e quasi ogni giorno si aggiungono agli archivi ufficiali nuove caratteristiche. È dunque indiscutibilmente più aggiornato della maggior parte degli altri sistemi di videoscrittura.

Leggenda: \LaTeX non è WYSIWYG

Dipende. Se per WYSIWYG (*What You See Is What You Get*, "ciò che vedi è ciò che ottieni") s'intende un software in grado di produrre nel documento finito testo e immagini disposti esattamente come si vede sullo schermo del calcolatore, \LaTeX è un programma WYSIWYG della migliore qualità.

Se invece per WYSIWYG s'intende un programma di videoscrittura in cui l'utente agisce direttamente sul testo già composto così come gli appare a

schermo e ogni sua azione si traduce in un'immediata variazione di quel testo, \LaTeX non è WYSIWYG ma WYSIWYM (*What You See Is What You Mean*, "ciò che vedi è ciò che intendi"), perché compone il documento in modo asincrono.

Leggenda: \LaTeX è troppo difficile

Questa frase si è sentita dire da fisici in grado di dividere gli atomi, da matematici che sanno dimostrare la trascendenza di π , da uomini d'affari che sanno leggere un foglio di bilancio, da storici che hanno compreso la politica bizantina e da linguisti che sanno decifrare la scrittura lineare B. \LaTeX non è immediato come i normali elaboratori di testo, certo, ma quel po' di studio e di pratica iniziali richiesto (possibilmente su documenti elementari) verranno presto ricompensati dalla qualità dei risultati.

Leggenda: \LaTeX è solo per matematici e scienziati

Niente affatto. Sebbene sia cresciuto nei campi della matematica e dell'informatica, due delle sue maggiori aree di espansione sono quelle umanistica ed economica, specie da quando ha preso piede l'XML, che ha sollevato nuove esigenze nell'ambito della tipocomposizione automatica.

1.2.5 Non tutte le strade portano a \LaTeX

Molti utenti, per i motivi più vari, vorrebbero beneficiare della potenza e delle qualità di \LaTeX senza pagare il piccolo prezzo dichiarato poco sopra, e si rivolgono ai vari programmi in circolazione che dichiarano di poter risolvere il problema " \LaTeX è difficile" rendendolo "facile" con interfacce grafiche intuitive e ricche pulsantiere per soddisfare ogni esigenza di scrittura (come non averci pensato!), e garantendo contemporaneamente l'eccellenza della composizione tipografica tipica di \TeX .

Gli autori di questa guida *scoraggiano con vigore* l'uso di quei programmi che tentano di nascondere \LaTeX all'utente mascherandolo da normale word processor. Il più accreditato di essi è LyX, ma è provato che appena gli si chiede qualcosa di non previsto dai suoi sviluppatori ci si trova in guai seri, e spesso la soluzione al problema si rivela più difficile di quanto non sarebbe stato usare \LaTeX fin dal principio. E lo stesso vale in genere per programmi analoghi. Per esempio, scrivere

```
\section{titolo}
```

non può essere davvero più complicato che premere un pulsante Sezione.

1.2.6 \LaTeX : errori da evitare

S'intendano i punti seguenti come un promemoria dei più comuni errori tipici dell'utente alle prime armi e se ne faccia tesoro.

1. *Non leggere con attenzione una buona guida di base.*
Non si può pensare di cominciare a usare \LaTeX senza affiancargli un manuale. Le guide sono scritte per essere lette e, se ben fatte, permettono di evitare gran parte degli errori iniziali.
2. *Installare una distribuzione di \LaTeX non completa.*
Sebbene possa sembrare inutile occupare quasi 4 GiB di spazio sul disco per installare un sistema \TeX completo, sapendo bene che non lo si sfrutterà per intero praticamente mai, tuttavia si consiglia di farlo ugualmente: ci si mette al riparo da eventuali problemi e si è certi

di avere a disposizione tutti i pacchetti che servono. La capienza dei moderni dischi fissi non dovrebbe preoccupare in questo senso.

3. *Lavorare con la distribuzione non aggiornata.*
È sempre bene aggiornare la distribuzione periodicamente: gli aggiornamenti non aggiungono solo nuove funzioni a pacchetti e a classi, ma risolvono anche eventuali malfunzionamenti lamentati dagli utenti. L'unica circostanza in cui potrebbe essere sconsigliabile farlo è quando si sta ultimando la revisione un documento importante: le migliorie potrebbero disattivare alcune funzioni fondamentali per il documento e ci potrebbe non essere abbastanza tempo per studiare quelle nuove e apportare al sorgente i cambiamenti richiesti. In tal caso, meglio attendere di aver terminato il lavoro.
4. *Pretendere di scrivere un intero sorgente unicamente premendo pulsanti con il mouse.*
Anche se le ricche pulsantiere che alcuni editor offrono per le varie funzioni *potrebbe* far sembrare più semplice scrivere codice \LaTeX , tuttavia si sconsiglia fortemente di abituarcisi. Si scriva il codice sempre a mano imparandone il più possibile: non si fa più fatica ed è enormemente più utile: sarà possibile lavorare senza problemi anche su editor diversi dal proprio.
5. *Installare su una distribuzione di Linux basata su Debian la \TeX Live approvata dal consorzio omonimo.*
Per motivi interni al consorzio Debian, la \TeX Live ufficialmente disponibile è sempre obsoleta di almeno due anni. Si risolve il problema seguendo la procedura indicata nella sezione 2.4 a pagina 16.
6. *Pretendere, come primo documento, di riprodurre layout e caratteri del proprio libro preferito o quelli di altri word processor, riempiendo il sorgente di molte e avanzate personalizzazioni.*
Questo è forse l'errore più tipico del principiante. Come si scoprirà nelle prossime pagine, \LaTeX è *molto* diverso dai soliti programmi di videoscrittura, e la cosa migliore è sempre affidarsi a classi e pacchetti che propongano layout e combinazioni di font predefinite tra cui scegliere. Fare pasticci in questo senso è più facile di quel che sembra, e usare tutti i font con la stessa facilità di altri programmi non è così semplice: le personalizzazioni sono pane per i denti dell'utente avanzato.
7. *Non cercare la risposta alla domanda «come si fa?» innanzitutto nella documentazione dei pacchetti.*
Ammesso di aver individuato il pacchetto che fa al caso proprio, la sua documentazione (quasi sempre in inglese) spiega tutto quello che c'è da sapere per ottenere ciò che serve. Se non la si legge, delle due l'una: o non si sa l'inglese o si è troppo pigri. Nei (pochi) casi in cui la documentazione risulti criptica, si può chiedere aiuto pubblicamente al forum del \CTAN , in modo che la soluzione al problema rimanga visibile a tutti.

2

INSTALLARE E AGGIORNARE

Questo capitolo spiega come procurarsi tutto l'occorrente per usare \LaTeX , come installarlo nel proprio calcolatore e come aggiornarne la distribuzione (che in questa guida si considererà *sempre* nell'ultima versione disponibile). I programmi descritti sono gratuiti e, dove non altrimenti indicato, disponibili per tutti e tre i sistemi operativi considerati in questo capitolo:

- Windows (da XP in poi);
- Mac OS X (dalla versione 10.5 Leopard in poi);
- Linux (nelle distribuzioni indicate nel paragrafo [2.4](#) a pagina [16](#)).

2.1 FERRI DEL MESTIERE

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

1. un editor di testi con cui scrivere il file sorgente;
2. il programma \LaTeX , che lo elabora e produce il documento tipocomposto;
3. un programma per visualizzare il documento finito.

Si eviti da subito l'errore, molto frequente all'inizio, di confondere \LaTeX (un "motore" del tutto invisibile all'utente) con l'editor (ciò che effettivamente appare sullo schermo): i due programmi sono completamente *indipendenti*, tanto che in teoria si può usare un editor qualunque, da quelli più elementari già presenti nel proprio computer a quelli più complessi capaci di gestire numerosi linguaggi di programmazione.

Altrettanto spesso si confondono i due programmi `latex` e `pdflatex`: il primo produce file `.dvi` (oggi indispensabili in circostanze così limitate da non meritare una trattazione in questa guida), il secondo file `.pdf`. La confusione nasce perché il nome \LaTeX viene usato indifferentemente per indicare entrambi, e in qualche editor si chiama così anche il pulsante che avvia la composizione.

In questa guida con \LaTeX s'intende *sempre* il programma `pdflatex`, e con le espressioni *documento composto* e *documento finito* un file in formato PDF.

2.1.1 Editor e visualizzatore: \TeX works

Anche se oggi molti software di videoscrittura tradizionali prevedono estensioni per trasformare il testo immesso in codice \LaTeX , il consiglio per chi muove i primi passi con questo programma è di usare senz'altro un editor *dedicato*, cioè destinato esclusivamente a \LaTeX e ideato per facilitarne il più possibile l'interazione con l'utente: i vantaggi sono molti e sostanziali. I più diffusi tra essi, inoltre, comprendono anche un visualizzatore di PDF, eliminando così i possibili problemi derivanti dall'usarne uno esterno.

Tabella 2: Percorsi di T_EX nei diversi sistemi operativi.

Sistema operativo	Percorso di T _E X
Windows	C:\Programmi\texlive\2012\bin\win32
Mac OS X	/usr/texbin
Linux	/opt/texbin

Gli editor che gestiscono codice L^AT_EX sono moltissimi, come si può scoprire con una veloce ricerca in Rete. Tra tutti vanno segnalati almeno:

- TeXShop (☞ **TEXSHOP**), nativo di Mac OS X;
- Kile (☞ **KILE**), nativo di Linux;
- Texmaker (☞ **TEXMAKER**) e TeXstudio (☞ **TEXSTUDIO**), programmi multipiattaforma con caratteristiche molto simili a quelle di T_EXworks, che si descrive nel prossimo paragrafo;
- Emacs (☞ **EMACS**) nelle sue diverse incarnazioni.

Si ricordi che la scelta finale è anche una questione di gusto.

Quello raccomandato in questa guida è T_EXworks (☞ **TEXWORKS**), un semplice ambiente di lavoro per L^AT_EX molto adatto ai principianti perché permette di evitare facilmente i tipici errori dell’inizio. Semplicità d’uso e immediatezza ne hanno fatto l’editor “ufficiale” di T_EX Live (si veda il paragrafo 2.1.2 a pagina 14): il programma, perciò, si troverà *già installato* sulla macchina (tranne che su Linux). Si noti che sarà sempre possibile, se lo si desidera, usare l’ultima versione al posto di quella preinstallata.

Caratteristiche del programma

Rinunciando alle tipiche pulsantiere degli editor convenzionali, l’interfaccia grafica di T_EXworks è semplice e intuitiva e privilegia le poche funzioni davvero importanti: se ne può vedere un esempio nella figura 4 a pagina 14.

Il programma gestisce numerosi motori di composizione e permette l’interazione con l’utente (si veda il paragrafo 3.1 a pagina 18). Offre le normali funzioni di cercare e sostituire e permette di aggiungere nuovi motori di composizione a quelli previsti. Infine, T_EXworks è compatibile con la tecnologia SyncT_EX: sorgente e antepagina cioè, sono *sincronizzati*, e si può “saltare” facilmente da un punto dell’uno al corrispondente punto dell’altro (e viceversa) risparmiando molto tempo. Si noti che attualmente c’è una limitazione, infatti non si può stampare il PDF direttamente dal visualizzatore: si risolve il problema usando un altro programma.

Di seguito si elencano le istruzioni fondamentali per il corretto funzionamento di T_EXworks:

- nel primo riquadro da sinistra nella parte bassa della finestra del sorgente si lasci LF, nel secondo UTF-8;
- si controlli che la prima voce della lista Percorso di T_EX e relativi programmi nel pannello TeXworks → Preferenze... → Composizione sia quella indicata nella tabella 2;
- s’installino i dizionari italiani di OpenOffice.org se si desidera un semplice controllo ortografico durante la scrittura.

Commenti speciali e gestione di documenti in più file

Un grande pregio che \TeX works condivide con pochissimi altri editor (solo \TeX Shop, \TeX studio, Emacs e forse Vim permettono qualcosa di simile) è la capacità di interpretare i *commenti speciali*, cioè righe che cominciano con la stringa `% !` da scrivere *come prima cosa* nel sorgente e che, come i commenti veri e propri (si veda il paragrafo 3.4.5 a pagina 28), vengono ignorati da \LaTeX ma *non* dall'editor, che in modo automatico si autoconfigura corrispondentemente. Non sono obbligatori, ma sono una buona abitudine: fissando *una volta per tutte* le impostazioni più importanti del documento e avendo la precedenza sulle impostazioni dell'editor, evitano all'utente i problemi descritti nel paragrafo 3.2.1 a pagina 19 e in [Beccari e Gordini, 2012].

Eccone la sintassi generale riconosciuta da \TeX works, \TeX Shop e \TeX studio (ma non da Emacs, alla cui documentazione si rimanda):

```
% ! $\text{\TeX}$  <parola chiave> = <valore>
```

dove:

- *<parola chiave>* è una delle quattro voci spiegate più sotto;
- *<valore>* dipende dalla *<parola chiave>*.

Si noti che maiuscole, minuscole e spazi, qui evidenziati con il segno `_`, vanno *sempre* rispettati (\TeX Shop permette di inserire queste righe anche automaticamente con appositi pulsanti).

Le possibili parole chiave sono:

encoding dichiara la codifica con cui è scritto il sorgente: si metta sempre UTF-8 (verificando di aver caricato *anche* `inputenc` nello stesso modo, si veda il paragrafo 3.2.1 a pagina 19) e non si avranno problemi;

program dichiara il motore di composizione, che di solito è `pdflatex` o `xelatex` (in tutte minuscole);

root dichiara la condizione di file principale e di file secondario nei documenti suddivisi in più file (si veda poco sotto);

spellcheck attiva il controllo ortografico della lingua del documento: per l'italiano si scriva `it-IT` (dopo aver installato i dizionari italiani).

\TeX works gestisce facilmente anche progetti suddivisi in più file come una tesi o un libro (si veda il paragrafo 3.14 a pagina 44) permettendo di comporre comodamente il documento dalla finestra in cui ci si trova anziché tornare ogni volta al file principale. Immaginando di lavorare a questa guida, il cui file principale si chiama `artelatex.tex`, basta scrivere

```
% ! $\text{\TeX}$  root = artelatex.tex
```

in testa a ciascun file da includervi se file principale e secondario stanno nella stessa cartella, oppure

```
% ! $\text{\TeX}$  root = ../artelatex.tex
```

se quest'ultimo è in una sottocartella.

Infine si dà il blocco completo dei commenti speciali con la sintassi di \TeX Shop per un documento in italiano da comporre con \LaTeX o \XeLaTeX (si veda il paragrafo B.2.2 a pagina 204):

```
% ! $\text{\TeX}$  encoding = UTF-8 Unicode
% ! $\text{\TeX}$  TS-program = pdflatex
% ! $\text{\TeX}$  TS-program = xelatex
% ! $\text{\TeX}$  spellcheck = it-IT
% ! $\text{\TeX}$  root = <nome del file principale>.tex
```

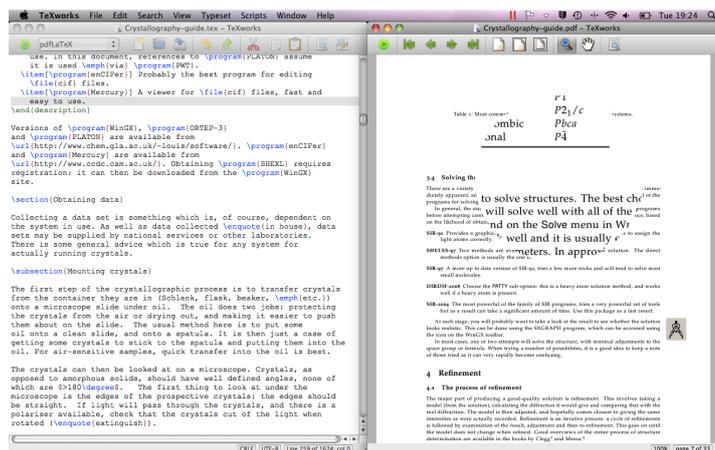


Figura 4: Una tipica schermata di TeXworks.

Si noti che:

- se si compone con Xe_{La}TeX il primo e il secondo commento vanno omessi;
- se si compone con L^ATeX va omesso il terzo;
- se il documento è in un solo file, l'ultimo commento non serve.

Per motivi di spazio, i commenti speciali *saranno omessi* in tutti i codici di questa guida.

2.1.2 TeX Live

L^ATeX è uno, ma prende forma in differenti versioni (non così tante) che si chiamano *distribuzioni*. Una distribuzione è una raccolta di file e altro software (il programma L^ATeX vero e proprio, uno o più editor dedicati e di solito altri programmi accessori) autosufficiente per produrre un documento finito. Si può installare direttamente da Internet, dal disco rigido dopo averla scaricata oppure da DVD.

Il luogo di riferimento nel Web da cui scaricare il materiale ufficiale su L^ATeX è CTAN (*Comprehensive T_EX Archive Network*, “rete di archivi completi per T_EX”, , CTAN), una rete di server dislocati in tutto il mondo, uguali tra loro e ciascuno contenente una copia integrale del sito originale (si chiamano anche *mirror*, “specchio”): ci si può così servire dal mirror più vicino, evitando di sovraccaricare la Rete e abbreviando i tempi dell’operazione.

In questa guida si consiglia di installare TeX Live (, TEXLIVE): è una distribuzione affidabile, mantenuta da decine di sviluppatori e aggiornata annualmente (quella corrente è TeX Live 2012). Gli utenti di Windows possono contare sull’alternativa di MiKTeX (, MIKTEX), che gira solo su quel sistema operativo. Qualunque delle due si scelga, si abbia cura di farne un’installazione *completa*: in caso contrario, potrebbero presentarsi alcuni fastidiosi problemi (si veda la sezione 3.2.1 a pagina 19).

TeX Live si modifica nel tempo, perché quasi ogni giorno molti dei *pacchetti* che la compongono (si veda il paragrafo 3.12 a pagina 41) vengono perfezionati, altri di nuovi vengono aggiunti agli archivi in Rete e altri ancora rimossi. Quelli già presenti riescono a soddisfare praticamente tutte le esigenze di scrittura, ma per mantenere la propria distribuzione sempre efficiente bisogna *aggiornarla* alle ultime versioni dei pacchetti e *installare* quelli nuovi continuamente creati e messi a disposizione.

Nei paragrafi seguenti si spiegherà:

- come installare T_EX Live sui diversi sistemi operativi (si richiedono dai 3 ai 4 GiB di spazio sul disco rigido);
- come usare il programma `tlmgr` (*TeX Live ManaGeR*) per aggiornarla e aggiungervi automaticamente i nuovi pacchetti; si raccomanda di farlo senz'altro subito dopo aver installato la distribuzione e poi con una certa regolarità (una volta alla settimana potrebbe andare bene).

2.2 WINDOWS

2.2.1 Installare

La procedura descritta di seguito, da eseguire *per tutta la sua durata* con una connessione a Internet attiva, permette di installare sul proprio calcolatore T_EX Live *aggiornata* alle ultime versioni dei pacchetti.

1. Si scarichi l'archivio compresso `install-tl.zip` da
<http://www.tug.org/texlive/acquire-netinstall.html>
 e lo si decomprima.
2. Si apra la cartella risultante `install-tl-2012mmgg` (la data esatta è di solito quella corrente), si esegua *come amministratore* il file batch `install-tl` al suo interno e si attenda finché non compare l'omonima finestra `Install-tl`. (Si noti che Windows Vista e 7 richiedono *esplicitamente* i privilegi di amministratore: si clicchi sul file con il pulsante destro e si scelga *Esegui come amministratore* nel menù che compare.)
3. Si seguano le istruzioni sullo schermo senza cambiare nulla (si decida soltanto se installare o meno T_EX Live per tutti gli utenti), si prema il pulsante `Installa` e si attenda la fine dell'operazione.

Se la procedura appena descritta non dovesse andare a buon fine (tipicamente per imprevisti nella connessione), l'alternativa è la seguente.

1. Si scarichi l'immagine ISO `texlive2012.iso` da
<http://tug.org/texlive/Images/>
 e la si decomprima.
2. Si torni al punto 2 della procedura precedente e la si segua dall'esecuzione del file batch alla fine.

Si noti che la T_EX Live installata con questo secondo metodo *non è aggiornata*.

2.2.2 Aggiornare

L'interfaccia grafica di `tlmgr` su Windows è T_EX Live Manager. Di seguito se ne dà la procedura per la prima volta. Per gli aggiornamenti successivi si salti il punto 2.

1. Si avvii il programma (Start → Programmi → TeX Live 2012 → TeX Live Manager).

2. Si apra la finestra Carica archivio pacchetti (tlmgr → Carica altro archivio...) e ci si assicuri che nella casella Archivio pacchetti: ci sia scritto

<http://ctan.mirror.garr.it/mirrors/CTAN/systems/texlive/tlnet>

In caso contrario, lo si scriva o lo si scelga premendo il pulsante Archivio dei pacchetti remoto predefinito e si prema Ok.

3. Si prema il pulsante Aggiorna installati in basso, si attenda la fine dell'aggiornamento e si chiuda il programma.

2.3 MAC OS X

2.3.1 Installare

Agli utenti di Mac OS X si raccomanda di installare T_EX Live tramite MacT_EX (☞ **MACTEX**), un installer che provvede a tutto il necessario. Segue la procedura.

1. Si scarichi l'archivio compresso MacTeX.pkg da

<http://www.tug.org/mactex/2012/>

2. Lo si esegua e se ne seguano le istruzioni fino a completare l'installazione.

2.3.2 Aggiornare

Su Mac OS X, T_EX Live appena installata *non* è aggiornata alle ultime versioni dei pacchetti: va perciò aggiornata attraverso T_EX Live Utility, l'interfaccia grafica di tlmgr per Mac. Di seguito si dà la procedura per la prima volta. Per gli aggiornamenti successivi si salti il punto 2.

1. Si avvii il programma (Applicazioni → TeX → TeX Live Utility).

2. Ci si assicuri che nella barra degli indirizzi in alto ci sia scritto

<http://ctan.mirror.garr.it/mirrors/CTAN/systems/texlive/tlnet>

In caso contrario, lo si scriva o lo si scelga dai mirror elencati alla voce Italy nella finestra TeX Live Repositories (Configure → Manage Repositories → EUROPE), si chiuda il programma e lo si riavvii.

3. Al termine della ricerca dei pacchetti, che il programma effettua automaticamente, si scelga la voce Update All Packages dal menù Actions (o si personalizzi la barra degli strumenti facendoci comparire il corrispondente pulsante Update All), si attenda la fine dell'aggiornamento e si chiuda il programma.

2.4 LINUX

A causa di alcuni limiti imposti dagli sviluppatori, non si può installare e aggiornare T_EX Live sulle distribuzioni Linux con la semplicità degli altri sistemi operativi. Le istruzioni descritte nella guida di [Gregorio, 2012] però, permettono di farlo con estrema tranquillità. La procedura lì descritta, cui si rimanda, funziona su una distribuzione di Ubuntu correttamente installata e, con qualche modifica, anche su Fedora (e simili) e OpenSuSe (e simili).

3 | BASI

3.1 PER COMINCIARE

In questo paragrafo si mostrano le semplici fasi per realizzare un documento con \LaTeX , dalla scrittura del sorgente alla stampa.

Scrivere il codice

Si crei una cartella nella quale mettere *tutti* i file del documento. Dopo di che, con l'editor scelto si apra un nuovo file e si scriva il seguente codice (tutte le istruzioni verranno spiegate in questo e nel successivo capitolo):

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

\begin{document}
Ecco il mio primo documento con  $\text{\LaTeX}$ .
\end{document}
```

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

Comporre

Si componga il codice premendo l'apposito pulsante. Via via che \LaTeX elabora il sorgente, ne rendiconta dettagliatamente la composizione in un file apposito con estensione .log (è il *log*, in gergo) conservato in una cartella del sistema, e contemporaneamente ne mostra sullo schermo una versione ridotta, più o meno come la seguente (per brevità se ne sono omesse alcune parti, segnalate con ...):

```
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012) ...
./primo.tex ...
LaTeX2e <2011/06/27>
Babel <v3.8m> and hyphenation patterns for english, ... italian, loaded.
(/usr/local/texlive/2012/texmf-dist/tex/latex/base/article.cls
Document Class: article 2007/10/19 v1.4h Standard LaTeX document class)
(/usr/local/texlive/2012/texmf-dist/tex/latex/base/fontenc.sty
(/usr/local/texlive/2012/texmf-dist/tex/latex/base/tlenc.def))
(/usr/local/texlive/2012/texmf-dist/tex/latex/base/inputenc.sty
(/usr/local/texlive/2012/texmf-dist/tex/latex/base/utf8.def
(/usr/local/texlive/2012/texmf-dist/tex/latex/base/tlenc.dfu)))
(/usr/local/texlive/2012/texmf-dist/tex/generic/babel/babel.sty
(/usr/local/texlive/2012/texmf-dist/tex/generic/babel/italian.ldf))
Output written on primo.pdf (1 page, 15596 bytes).
SyncTeX written on primo.synctex.gz.
Transcript written on primo.log.
```

Si noti che premere il pulsante di composizione dell'editor equivale a eseguire dalla riga di comando l'istruzione

Tabella 3: Modi di interazione tra L^AT_EX e utente.

Chiave	Ordina a L ^A T _E X di
Invio	ignorare l'errore e riprendere la composizione (va dato a ogni successivo errore)
q	continuare la composizione senza visualizzarne le informazioni e arrestarla solo quando un file non viene trovato (<i>carry on quietly</i>)
r	fare come nel caso precedente, ma informando l'utente sulla composizione (<i>run</i>)
s	ignorare gli errori e arrestare la composizione solo quando un file non viene trovato (<i>scroll</i>)
e	arrestare la composizione mostrando il documento composto fino a quel punto e tornare all'editor con il cursore nella riga che contiene l'errore (<i>edit</i>)
h	soccorrere l'utente con un messaggio d'aiuto (<i>help</i>)
i	attendere dall'utente la stringa corretta senza modificare il sorgente (<i>insert</i>)
x	arrestare immediatamente la composizione (<i>exit</i>)

```
pdflatex primo.tex
```

L'interfaccia a riga di comando (che su Windows si chiama *Prompt dei comandi*, su Mac *Terminale* e su Linux *Terminale* o *Console*) si avvia con modalità proprie di ogni sistema operativo.

In questo modo, si è consapevoli del fatto che è L^AT_EX (e non l'editor) il programma che elabora il codice `.tex`, che emette i messaggi e che produce il PDF, lavorando "dietro le quinte". La composizione non è dunque semplicemente "il tempo che bisogna attendere per vedere il PDF", ma il processo durante il quale il programma comprende le intenzioni dell'utente (espresse con i comandi), e le trasforma in un file (tipo)grafico.

Errori e warning

Per quanto scrupolosi si possa essere, di tanto in tanto qualche errore s'infila nel sorgente. Quando L^AT_EX s'imbatte in qualcosa che non capisce o che non può fare, arresta la composizione in corso e mostra quello che non va in un messaggio, che può essere di due tipi:

- un messaggio di vero e proprio *errore* (nomi di comandi scritti scorrettamente, parentesi dimenticate, caratteri speciali usati per sbaglio, per esempio): la composizione si arresta e il programma rimane in attesa di istruzioni da parte dell'utente;
- un messaggio di *avviso* (o *warning*, in gergo), meno grave del primo (righe troppo lunghe o troppo corte, pagine riempite troppo o troppo poco, riferimenti incrociati irrisolti, per esempio): il programma semplicemente lo notifica ma porta a termine la composizione.

Gli errori vanno corretti *obbligatoriamente*. In caso contrario, se sono pochi e non troppo importanti si potrà comunque completare la composizione, anche se in genere con un risultato diverso da quello atteso; se invece sono gravi (ne basta anche uno solo), la composizione potrebbe bloccarsi del tutto. Ai warning, semplici commenti su questioni di secondaria importanza, ci si potrà dedicare a lavoro ultimato.

Un esempio chiarirà le idee. Il nome di un comando scritto scorrettamente provoca l'arresto della composizione e la comparsa di un messaggio di questa forma (uguale per tutti i messaggi d'errore di questo tipo):

```
! Undefined control sequence.
l.6 Ecco il mio primo documento con \latex

?
```

Quando trova un errore, il programma segnala all'utente:

- la natura del messaggio (un errore, in questo caso) e il suo emittente (TeX, in questo caso: un errore emesso da L^AT_EX comincia sempre con ! LaTeX Error:),
- la natura dell'errore (in questo caso, Undefined control sequence, "sequenza di controllo non definita", ovvero "comando sconosciuto");
- la riga (*line*) esatta del codice sorgente in cui si trova l'errore (l.6);
- il testo contenuto nella riga appena segnalata (o parte di esso), alla fine del quale si trova l'errore in questione: se il testo continuasse, la riga incriminata verrebbe spezzata in corrispondenza del punto problematico e continuata subito sotto.

Come comportarsi? Se si sta usando un editor interattivo (come tutti quelli consigliati in questa guida), sarà possibile comunicare a L^AT_EX le proprie intenzioni in uno dei modi mostrati nella tabella 3 nella pagina precedente, dando poi Invio. A essere precisi, un errore come quello appena mostrato può avere due cause: o è sbagliato il nome del comando (quello giusto è \LaTeX e non \latex: attenzione alle maiuscole!); oppure, sebbene scritto correttamente, si sta usando un comando definito da un pacchetto non caricato. Essendo giusta la prima ipotesi, basta dare x e Invio, correggere il codice, registrare il file e ricomporre (oppure, con alcuni editor, ricomporre direttamente).

Si noti che, per quanto generalmente molto chiari, i messaggi di L^AT_EX potrebbero richiedere un po' d'esercizio iniziale per essere interpretati correttamente a prima vista. In particolare, si ricordi che le notifiche d'errore possono provenire da più d'una fonte (TeX, L^AT_EX, la classe in uso, uno dei pacchetti), che le cause per uno stesso errore possono essere più d'una e che la riga indicata si riferisce al sorgente *in quel momento* elaborato: lo si tenga presente se il proprio documento è suddiviso in più file. A fronte di errori ricorrenti e apparentemente inspiegabili, infine, un metodo spesso efficace è quello di eliminare i file ausiliari generati dalla composizione e ricomporre il documento.

Per una panoramica su errori più frequenti e possibili soluzioni, si vedano [Lamport, 1994] e [Goossens *et al.*, 2004].

Visualizzare e stampare

Se si usa un editor dedicato e il sorgente non contiene (più) errori, la composizione andrà a buon fine e il visualizzatore di PDF si attiverà automaticamente mostrando il documento finito, che si potrà poi stampare con le modalità proprie del programma.

3.2 CODIFICHE E LINGUE

3.2.1 Le codifiche di L^AT_EX

L^AT_EX è un programma nato per scrivere documenti *in inglese* ad alto contenuto matematico, e a questo scopo è stato originariamente corredato di

una dotazione minima di caratteri (lettere, numeri e pochi altri segni) del tutto sufficiente, tant'è vero che il codice

```
\documentclass{<...>}
\begin{document}
...
\end{document}
```

permette di scrivere senza problemi un documento in quella lingua. La corretta scrittura di parole straniere, tuttavia, potrebbe richiedere di caricare i pacchetti descritti in questo paragrafo anche in un documento completamente in inglese.

Com'è noto, l'inglese si scrive senza accenti o caratteri particolari, contrariamente alla vasta maggioranza delle altre lingue che usano l'alfabeto latino. Gli unici caratteri "complicati" dell'italiano sono le vocali accentate, presenti (non proprio tutte) anche su una tastiera italiana. Affinché \LaTeX le "accetti" se inserite nell'editor con i tasti appositi, restituendole adeguatamente nel documento composto e sillabando correttamente le parole che le contengono, *immediatamente dopo* la dichiarazione di classe vanno caricati altri due pacchetti nell'ordine seguente:

```
\documentclass[<...>]{<...>}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
```

dove:

- `fontenc` (*font encoding*, "codifica dei font") si occupa dei font che si vedranno nel documento composto e fornisce a \LaTeX i caratteri particolari di una certa lingua già disegnati. Si noti che per funzionare al meglio il pacchetto richiede una distribuzione *completa* di \LaTeX , altrimenti i caratteri potrebbero apparire "sgranati" sullo schermo (ma non a stampa).
- `T1` è la *codifica di output*, in gergo, per scrivere in italiano e in molte altre lingue occidentali: per lingue o alfabeti particolari si usano altre codifiche. In documento multilingue l'*ultima* codifica dichiarata è quella della lingua principale: `[T2A, T1]` per un documento in italiano e russo, per esempio.
- `inputenc` (*input encoding*, "codifica di input") serve a \LaTeX per interpretare correttamente i caratteri immessi nell'editor.
- `utf8` è la *codifica di input*, in gergo, che permette di scrivere nell'editor i segni di numerosi alfabeti direttamente dalla tastiera, evitando di dover caricare ogni volta la codifica adatta alla lingua del documento. Se servissero caratteri non compresi neppure in `utf8` si può ampliare il campo con la codifica `utf8x`, da sola o abbinata al pacchetto `ucs` (da caricare *prima* di `inputenc`). Se il proprio editor non supporta *pienamente* `utf8`, si usi `latin1`.

Non ci sono alternative: senza i due pacchetti appena descritti le vocali accentate non vengono visualizzate nel documento composto. Caricandoli, inoltre, il vecchio metodo di comporre *a mano* i caratteri con l'accento e fonte di non pochi problemi di sillabazione nelle parole che li contenevano, mostrato nell'esempio seguente,

```
Basta! Non se ne pu\`o pi\`u!
Perch\`e non c'\`e piet\`a
per chi deve scrivere cos\`i?
```

Basta! Non se ne può più! Perché non c'è pietà per chi deve scrivere così?

non serve più, e il codice è decisamente più pulito:

```
Però: caricando il pacchetto
giusto la vita diventerà così
facile che c'è molto più gusto!
```

```
Però: caricando il pacchetto giusto la
vita diventerà così facile che c'è molto
più gusto!
```

Si noti che, a patto di usare la tastiera “giusta” e $\text{X}_{\text{L}}\text{A}\text{T}\text{E}\text{X}$ (si veda il paragrafo [B.2.2](#) a pagina 204), la codifica Unicode permette di scrivere direttamente nel codice anche caratteri di alfabeti non latini.

3.2.2 Problemi con le codifiche

Un ulteriore aspetto da tenere bene a mente, spesso fonte di grattacapi e danni a volte irrimediabili, è il rapporto tra la codifica con cui è scritto un sorgente e quella con cui è impostato l’editor in uso, che *devono* coincidere, altrimenti aprendo un file sconosciuto si potrebbero vedere caratteri bizzarri sullo schermo. Se ciò si verificasse, *non* si componga né si registri il file *per nessun motivo*, ma lo si chiuda e si seguano le procedure descritte in [Beccari e Gordini, 2012], a cui si rimanda per individuare e risolvere i principali problemi in questo senso. Comporre il file, infatti, potrebbe danneggiarlo più o meno seriamente. In ogni caso, si legga la documentazione del proprio editor: potrebbe rivelarsi molto utile per evitare di questi pasticci.

3.2.3 $\text{L}_{\text{A}}\text{T}\text{E}\text{X}$ e le lingue

Il terzo e ultimo pacchetto da caricare *sempre* è `babel`, che agisce su parole fisse (cioè le voci generate automaticamente dai comandi raccolti nella tabella 57), date, convenzioni tipografiche e sulla scelta delle regole di sillabazione, che per italiano e principali lingue europee sono già comprese in $\text{T}_{\text{E}}\text{X}$ Live. Come opzioni prende una o più lingue e per un documento in italiano lo si carica nel modo seguente:

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

dove l’*ultima* dichiarata è la lingua principale del documento. Inoltre, per ciascuna lingua definisce nuovi comandi per semplificare l’immissione dei caratteri particolari nazionali, come si può vedere nella documentazione del pacchetto o in [Gregorio, 2010].

Si dà, infine, il tipico inizio di un sorgente per un documento in italiano con la corretta sequenza dei pacchetti da caricare:

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

Il pacchetto `babel` definisce alcuni comandi molto utili per trattare correttamente ciascuna lingua in un documento multilingue. Supponendo di dover scrivere un documento in italiano con alcune parti in inglese, `babel` andrà caricato così:

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

Per singole parole o brevi frasi in lingua straniera è disponibile il comando

```
\foreignlanguage{lingua}{testo}
```

Tabella 4: Principali file ausiliari di \LaTeX .

Prodotti da	Estensione	Descrizione
Utente	.bib .jpg, .pdf, .png .tex	Database bibliografico Formati grafici per \LaTeX File sorgente
Classi, pacchetti e stili	.bst .cls .sty	Stile bibliografico per Biber Classe di documento Pacchetto
Composizione	.aux .lof .log .lot .toc	Trasporta informazioni generiche Indice delle figure Rendiconta l'ultima composizione Indice delle tabelle Indice generale
Pacchetti e programmi	.bbl .blg .idx .ind .ilg .out	Bibliografia creata con Biber Rendiconto di Biber Voci dell'indice analitico Prodotto di MakeIndex Rendiconto di MakeIndex Segnalibri ipertestuali
Output	.pdf	Prodotto di $\text{PDF}\LaTeX$

Per porzioni di testo in lingua più consistenti è disponibile l'ambiente

```
\begin{otherlanguage*}{\langlelingua\rangle}
...
\end{otherlanguage*}
```

che però non traduce parole fisse e date. Se invece se ne avesse bisogno, si usi l'analogo ambiente `otherlanguage`.

3.3 FILE CON CUI SI HA A CHE FARE

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

La tabella 4 ne raccoglie e descrive i principali.

3.4 FILE SORGENTE

Un sorgente di \LaTeX è un file di puro testo che contiene sia il testo vero e proprio del documento sia i comandi che istruiscono \LaTeX su come trattarlo.

3.4.1 Comandi e ambienti

Un comune programma di videoscrittura a composizione sincrona e \LaTeX presentano per certi versi un funzionamento simile: entrambi ricevono dal-

Tabella 5: Principali tipi di comando in base a numero e tipo di argomenti richiesto. (F indica gli argomenti facoltativi, O quelli obbligatori.)

Argomenti		Esempio
F	O	
		<code>\LaTeX</code>
1		<code>\item[...]</code>
	1	<code>\emph{...}</code>
1	1	<code>\documentclass[...]{...}</code>
2	1	<code>\subfloat[...][...]{...}</code>
1	2	<code>\pdfbookmark[...]{...}{...}</code>
	3	<code>\multicolumn{...}{...}{...}</code>

L'utente sia il testo sia le istruzioni per impostarne l'aspetto. La differenza sostanziale, però, è che il primo le nasconde, proponendo all'utente menù da cui scegliere comandi "preconfezionati" che mostrano immediatamente i propri effetti; \LaTeX , invece, richiedendo di scrivere esplicitamente i comandi, mantiene queste istruzioni in superficie nei modi spiegati di seguito.

Comandi e dichiarazioni

Un *comando* è un'istruzione che ordina a \LaTeX di trattare in un certo modo una porzione più o meno ampia di testo. Si possono classificare i comandi di \LaTeX in base a forma e funzione.

Per quanto riguarda la *forma* si distinguono tre tipi di comando, a seconda che siano costituiti da:

- Un solo carattere non alfabetico. Questi comandi sono quattro in tutto: spazio `_`, tilde `~`, circonflesso `^` e trattino basso `_`.
- Una barra rovescia `\` seguita da un solo carattere non alfabetico (cioè non compreso fra A-Z o a-z). Un comando di questo tipo termina al primo carattere non alfabetico e uno o più spazi (che contano per un solo spazio) immediatamente dopo *non* vengono ignorati. Alcune tra le combinazioni più usate sono: `\{`, `\}`, `\%`, `\$`, `\-`, `\&`, `\#`, `\~`.
- Una barra rovescia `\` seguita da una sequenza di caratteri alfabetici. Un comando di questo tipo termina al primo carattere non alfabetico e uno o più spazi immediatamente dopo vengono ignorati. Si noti che questi comandi distinguono maiuscole e minuscole. Alcuni esempi: `\LaTeX`, `\emph`, `\documentclass`.

I comandi del terzo tipo che "producono testo" (come `\LaTeX`, `\TeX`, `\Ars`, `\dots`, `\today` e pochissimi altri) richiedono di essere "terminati", pena una spaziatura errata dopo di sé. La cosa migliore è usare la barra rovescia o un gruppo vuoto `{}`, come mostrano gli esempi seguenti:

<code>\Ars</code> è la rivista del Gruppo Utilizzatori Italiani di <code>\TeX</code> e <code>\LaTeX</code> . <code>\\[lex]</code>	<code>\ArsTeXnica</code> è la rivista del Gruppo Utilizzatori Italiani di <code>TeXe</code> \LaTeX .
<code>\Ars\</code> è la rivista del Gruppo Utilizzatori Italiani di <code>\TeX{}</code> e <code>\LaTeX</code> .	<code>\ArsTeXnica</code> è la rivista del Gruppo Utilizzatori Italiani di <code>TeX</code> e \LaTeX .

La scrittura *sempre* corretta di questi comandi è una di quelle contenute nel secondo esempio. Si noti che un qualunque segno di punteggiatura immediatamente dopo il comando elimina la necessità dello spazio esplicito.

Tabella 6: Caratteri speciali di \LaTeX .

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

Per quanto riguarda la *funzione* si distinguono due tipi di comando, a seconda della porzione di testo su cui hanno effetto:

- Comandi come `\textit{<testo>}` ordinano a \LaTeX di trattare in un certo modo solo il *<testo>* scritto tra le parentesi graffe.
- Comandi come `\itshape`, detti *dichiarazioni*, ordinano a \LaTeX di trattare in un certo modo tutto *tutto* il testo successivo al punto in cui vengono dati.

In altre parole, una dichiarazione è un comando che imposta uno o più aspetti generali della composizione, e può essere data:

- nel preambolo, e allora ha effetto sull'intero documento e si annulla soltanto con un'altra dichiarazione;
- nel corpo del documento, e allora va data in un *gruppo* (cioè una porzione di testo racchiusa di solito da parentesi graffe o comandi di inizio e fine ambiente).

Sono esempi di dichiarazione: `\small`, `\linespread`, `\appendix`.

Nell'esempio seguente si vedono all'opera alcuni comandi visti fin qui:

<pre>Data odierna: \today. \\ Sarò lì in \emph{dieci} minuti. \\ Tutto il {\itshape testo seguinte è in corsivo}.</pre>	<pre>Data odierna: 23 novembre 2012. Sarò lì in <i>dieci</i> minuti. Tutto il <i>testo seguente è in corsivo</i>.</pre>
---	---

Si noti che il comando `\today` produce la data in cui si compone il documento secondo le convenzioni della lingua in uso.

La giustapposizione degli elementi di un comando prende il nome di *sintassi* del comando. Ciò che va tra parentesi graffe si chiama *argomento obbligatorio*, mentre ciò che va tra parentesi quadre si chiama *argomento facoltativo*. Se gli elementi da scrivere nello stesso gruppo di parentesi sono più d'uno, vanno separati con una virgola *senza ulteriori spazi*. La tabella 5 nella pagina precedente mostra i principali tipi di comando in base a numero e tipo di argomenti che richiedono.

Ambienti

Un *ambiente* è una porzione di codice delimitata da un comando d'apertura e uno di chiusura, che \LaTeX tratta in un certo modo. La sintassi di un ambiente generico è:

Tabella 7: Scorciatoie da tastiera (italiana) per alcuni caratteri frequenti.

Carattere	Windows	Mac	Linux
‘	Alt + 96	⌥ 9	Alt Gr + ’
{	Alt + 123	⌥ ⇧ [Alt Gr + 7
}	Alt Gr + Maiusc + [⌥ ⇧]	Alt Gr + Shift + [
~	Alt + 125	⌥ ⇧]	Alt Gr + 8
	Alt Gr + Maiusc +]		Alt Gr + Shift +]
~	Alt + 126	⌥ 5	Alt Gr + ì

```
\begin{<ambiente>}[<...>]{<...>}
...
\end{<ambiente>}
```

dove:

- *<ambiente>* è il nome dell’ambiente;
- se presenti, argomenti facoltativi e obbligatori si scrivono dopo il solo comando d’apertura;
- l’ambiente va separato dal resto del testo con una riga bianca prima e dopo se ciò che contiene *non* appartiene al flusso del discorso (una figura, per esempio); non va separato in caso contrario.

L^AT_EX permette di annidare gli ambienti, purché l’ordine di chiamata venga rispettato:

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

3.4.2 Caratteri speciali

L^AT_EX interpreta in modo particolare alcuni caratteri molto richiesti nella scrittura del codice: sono i cosiddetti *caratteri speciali*, così chiamati perché non possono essere stampati normalmente se non come mostra la tabella 6 nella pagina precedente. La loro alta frequenza, però, si scontra con il fatto che su nessuna tastiera, a parte quella inglese internazionale, ci sono tutti, e perciò la loro scrittura richiede combinazioni di tasti o codici numerici particolari.

Si noti che:

- i caratteri { ‘ ~ } mancano sulla tastiera italiana: la tabella 7 indica le scorciatoie da prendere in questi casi (su Windows, il codice relativo va digitato *sul tastierino numerico*);
- si distingue con attenzione ‘ (virgoletta alta aperta, accento grave) da ’ (apostrofo, virgoletta alta chiusa, accento acuto);
- il comando `\textbackslash` *non* sostituisce la sequenza `\\`, come potrebbe sembrare: sono infatti due comandi distinti con distinte funzionalità (verranno considerati nei prossimi capitoli).

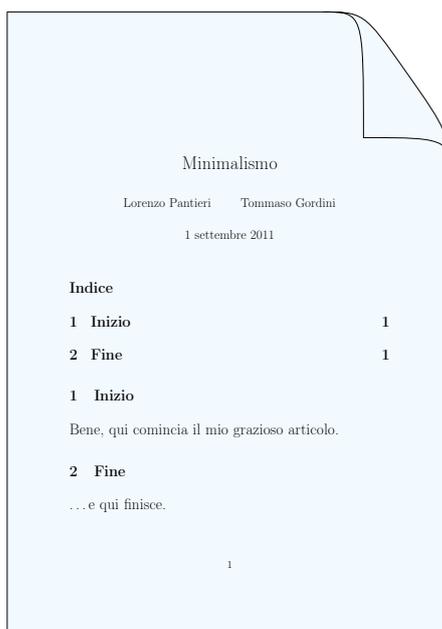


Figura 5: Elementare documento composto con \LaTeX .

3.4.3 Struttura del file sorgente

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

```
\documentclass{<...>}
```

e le dichiarazioni d’inizio e fine documento:

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

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

- il caricamento di pacchetti che estendono le capacità di \LaTeX ;
- le definizioni di comandi e ambienti personalizzati, che si consiglia di organizzare come indicato nel paragrafo 3.14 a pagina 44;
- le opzioni generali del documento.

Si noti che un sorgente \LaTeX richiede *un solo* preambolo.

Fra `\begin{document}` e `\end{document}` va scritto il *corpo del documento*, cioè il vero e proprio testo che \LaTeX elaborerà e mostrerà nel documento finito. La figura 5 mostra il risultato della composizione del codice seguente (si possono intercalare o meno righe bianche per evidenziarne la struttura e facilitarne l’individuazione delle parti):

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

\begin{document}
```

```

\author{Lorenzo Pantieri \and Tommaso Gordini}
\title{Minimalismo}
\maketitle

\tableofcontents

\section{Inizio}
Bene, qui comincia il nostro grazioso articolo\dots

\section{Fine}
\dots e qui finisce.

\end{document}

```

Dove:

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

\LaTeX ignora tutto ciò che si trova dopo `\end{document}`: questo spazio, perciò, potrebbe essere un buon posto per appuntare un promemoria sul documento in lavorazione.

3.4.4 Spazi e righe vuote

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

- una tabulazione è considerata come uno spazio;
- più spazi consecutivi sono considerati come un solo spazio;
- spazi o tabulazioni all’inizio di una riga vengono ignorati;
- una sola interruzione di riga è trattata come uno spazio;
- una riga vuota tra due righe di testo separa due capoversi;
- più righe vuote consecutive sono trattate come una sola riga vuota.

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

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

```
E neppure se si mettono
tanti spazi all'inizio di una
riga o se
la s'interrompe.
```

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

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

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

3.4.5 Commenti

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

Talvolta però va usato per spezzare parole troppo lunghe o per dividere righe in cui non sono permessi spazi bianchi o interruzioni (come nel caso di comandi troppo lunghi per stare in una sola riga dell'editor). Ecco all'opera:

```
Ecco un % semplice,
% ma istruttivo
esempio: Supercal%
          ifragilist%
          ichespiralidoso.
```

Ecco un esempio: Supercalifragilisti-chespiralidoso.

Per commenti più lunghi si può usare l'ambiente `comment` definito dall'omonimo pacchetto:

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

Ecco un altro esempio per includere commenti nel proprio documento.

3.4.6 Sorgenti ordinati

Spesso gli utenti di \LaTeX sottovalutano l'importanza di un sorgente "pulito" (con rientri, incolonnamenti, eccetera) e commentato. *Non* è indispensabile, ma si consiglia di farlo ugualmente: l'ordine ne facilita la gestione, specie se a uno stesso progetto lavorano più persone, e rende più facile individuare eventuali errori.

Durante la stesura si raccomanda di usare i commenti, di suddividere con chiarezza il documento e di aiutarsi eventualmente con rientri, incolonnamenti, a capo e righe vuote supplementari. Ulteriori consigli in questo senso verranno forniti nelle prossime pagine al momento opportuno.

Si realizzano ora le indicazioni precedenti in un sorgente “ben scritto”:

Esempio di articolo composto con L^AT_EX.

```
% Un articolo scritto con LaTeX
\documentclass[a4paper,11pt]{article}
\usepackage[T1]{fontenc}      % codifica dei font in uscita
\usepackage[utf8]{inputenc}  % lettere accentate da tastiera
\usepackage[italian]{babel}  % lingua principale del documento
\usepackage{lipsum}          % genera testo fittizio
\usepackage{url}             % per scrivere gli indirizzi Internet

\begin{document}

\author{Lorenzo Pantieri \and Tommaso Gordini}
\title{Il titolo}
\maketitle

\begin{abstract}
\lipsum[1]
\end{abstract}

\tableofcontents

\section{Un paragrafo}
\lipsum[1]

\subsection{Un sottoparagrafo}
\lipsum[1]

\section{Un paragrafo}
\label{sec:esempio}
\lipsum[1]

% Bibliografia
\begin{thebibliography}{9}
\bibitem{pantieri:arte}
Pantieri, Lorenzo e Tommaso Gordini (2012),
\emph{L'arte di scrivere con \LaTeX},
\url{http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf}.
\end{thebibliography}

\end{document}
```

3.5 CLASSI DI DOCUMENTO

La prima informazione che L^AT_EX si aspetta di trovare nel sorgente è il *tipo di documento* che si desidera realizzare (la *classe*, in gergo), da specificare *come primissima cosa* (seconda solo ai commenti speciali) con il comando

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

dove:

- *<opzioni>* sono le impostazioni generali del documento;
- *<classe>* è la classe di documento scelta.

Di seguito si elencano le principali classi di documento *standard* (cioè definite dal programma):

article per scrivere articoli;

report per scrivere relazioni o tesi suddivise in capitoli;

book per scrivere libri;

letter per scrivere lettere.

Esistono numerose altre classi non standard per i documenti più diversi. Tra quelle più diffuse ci sono memoir (che permette molta libertà nel personalizzare il documento), toptesi e suftesi (per tesi di laurea e dottorato), beamer (per presentazioni). Per esempio, questa guida è scritta con lo stile ClassicThesis basato sulle classi KOMA-Script (per maggiori dettagli si veda [Pantieri, 2007]).

Le *opzioni* date a `\documentclass` si dicono anche *globali*, perché agiscono sull'intero documento. Di seguito si descrivono quelle più comuni per le tre classi article, report e book (la classe letter meriterebbe una trattazione a parte, perciò non viene considerata). Si noti che alcune di esse hanno un'applicabilità limitata, come mostra la tabella 8 nella pagina successiva.

10pt, 11pt, 12pt Impostano la dimensione del font principale del documento. Omettendo l'opzione, il valore predefinito è 10pt.

a4paper, a5paper, ... Definiscono le dimensioni del foglio, che per impostazione predefinita è nel formato americano letterpaper. Le altre opzioni possibili sono executivepaper, legalpaper e b5paper.

oneside, twoside Specificano se verrà composto un documento a singola o doppia facciata rispettivamente. Per impostazione predefinita, le classi article e report sono a singola facciata e la classe book è a doppia facciata.

openany, openright L'opzione openany, predefinita nella classe report, fa cominciare un capitolo nella successiva pagina a disposizione; l'opzione openright, predefinita nella classe book, lo fa cominciare sempre in una pagina destra. Entrambe non sono disponibili nella classe article, che non ammette la suddivisione in capitoli.

twocolumn Dà a L^AT_EX le istruzioni per comporre l'intero documento su due colonne (si veda anche il paragrafo 11.3 a pagina 177).

titlepage, notitlepage Specificano se dopo il titolo del documento debba avere inizio una nuova pagina (come accade con report e book) o no (come accade con article) rispettivamente.

fleqn Allinea le formule a sinistra rispetto a un margine rientrato.

leqno Mette la numerazione delle formule a sinistra anziché a destra.

draft, final L'opzione draft evidenzia le righe composte in modo non ottimale con un rettangolino nero ■ accanto, facilitandone l'individuazione sulla pagina. Ciò non accade con final. Si noti che entrambe influenzano il comportamento degli altri pacchetti caricati o addirittura li disabilitano del tutto (si veda il paragrafo 12.1 a pagina 185 per risolvere il problema).

Un tipico sorgente, allora, potrebbe cominciare con la riga

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

che ordina a L^AT_EX di impaginare il documento come un articolo, su carta di formato A₄, con un carattere di 11 punti e impostato per la stampa fronte/retro.

Tabella 8: Opzioni più comuni delle classi standard di \LaTeX . I simboli ●, ◐ e ○ indicano rispettivamente che l'opzione è predefinita, applicabile (anche se non predefinita), non applicabile.

Opzione	book	report	article
10pt	●	●	●
letterpaper	●	●	●
oneside	◐	●	●
twoside	●	◐	◐
openany	◐	●	○
openright	●	◐	○
titlepage	●	●	◐
final	●	●	●

3.6 GESTIRE LA PAGINA

3.6.1 Il tormentone dei margini

I margini della pagina in tipografia rivestono funzioni importantissime, prima fra tutte quella di delimitare in modo chiaro il testo. Il lettore così potrà individuarlo agevolmente sulla pagina e appoggiare i pollici su uno spazio sufficientemente confortevole per maneggiare comodamente il documento [Bringhurst, 1992]. È perciò che nei documenti impostati per la stampa in fronte/retro i margini esterni sono più ampi di quelli interni, che appaiono duplicati perché adiacenti.

La maggior parte degli utenti europei, che stampa su carta in formato A4, ritiene troppo ampi i margini predefiniti da \LaTeX nelle classi standard, e che di conseguenza la pagina non sia sufficientemente riempita. Prima di buttarsi nella frenesia dell' "allarghiamo un po' questa strettissima pagina", però, è doveroso riflettere.

I margini di \LaTeX derivano da convenzioni tipografiche ampiamente verificate e accettate, e mettono l'utente nelle vantaggiose condizioni di potersene servire per ottenere risultati professionali *già alla prima composizione* e senza doverci mettere le mani. Modificarli, perciò, significherebbe dover studiare un (bel) po' di tipografia prima di raggiungere risultati accettabili.

L'esperienza dimostra che leggere diventa tanto più difficile quanto più numerosi sono i caratteri in una singola riga di testo: l'occhio è costretto a compiere movimenti più ampi e si affatica presto (perciò quotidiani e riviste sono stampati su più colonne). Robert Bringhurst ha codificato quest'esperienza nella sua celebre "regola", che considera ottimale il numero di circa 66 caratteri per riga (spazi inclusi), indipendentemente dal font usato [Pantieri, 2007]. L'ampiezza media in punti di un carattere corrisponde al rapporto tra la lunghezza dell'alfabeto latino minuscolo "abcdefghijklmnopqrstuvwxyz" e il numero di lettere che lo compongono (26). Se si considera che per riempire meglio la pagina \LaTeX usa già *in partenza* una riga più lunga del limite stabilito da Bringhurst, è chiaro che modifiche in questo senso vanno evitate il più possibile.

In alcune circostanze, tuttavia, può essere desiderabile o necessaria una maggiore copertura della pagina: tra i numerosi pacchetti scritti a questo scopo se ne consigliano due.

Il pacchetto LayAureo (se ne veda la documentazione in italiano) definisce un layout di pagina pronto per l'uso, permettendo di impostare facilmente anche lo spazio per la rilegatura con la chiave `binding=<dimensione>`. Il pacchetto agisce semplicemente caricandolo e non è personalizzabile: dunque, o piace o non piace.

Se servissero proporzioni di pagina ancora diverse (perché la propria facoltà impone un modello di tesi particolare, per esempio) da applicare anche a una sola pagina, può risolvere il problema il pacchetto `geometry`, completamente configurabile.

Si immagini di dover comporre un documento in formato A4 con margini superiore e inferiore di 3 cm, sinistro e destro (che nella stampa in fronte/re-ro diventano interno ed esterno) di 3,5 cm e di voler destinare alla rilegatura uno spazio di 5 mm. Il codice da scrivere nel preambolo è il seguente:

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

Tra le opzioni del pacchetto che come il precedente, si noti bene, agisce anche solo caricandolo, si consiglia sempre anche `heightrounded`, che modifica ulteriormente di poco le dimensioni della gabbia del testo per farle contenere un numero intero di righe.

Si eviti *nel modo più assoluto*, invece, di toccare comandi interni di \LaTeX come `\textwidth`, `\oddsidemargin`, eccetera, perché la loro azione non tiene in nessun conto le proporzioni di pagina [Fairbairns, 2012].

Talvolta il rilegatore potrebbe non sapere dove tagliare il foglio: il pacchetto `crop` stampa sul documento i crocini di taglio.

3.6.2 Interlinea e riempimento della pagina

Interlinea

Non pochi editori, relatori e regolamenti di facoltà impongono di impaginare pubblicazioni e tesi in modo non professionale, a partire dalla questione dell'*interlinea*. L'interlinea standard di \LaTeX (fissata a 1) garantisce un risultato tipografico ottimale e non andrebbe modificata senza una ragione precisa. Per farlo si consiglia il pacchetto `setspace`, che definisce tre interlinee *globali* da impostare *nel preambolo* come segue:

- `\singlespacing` (interlinea 1);
- `\onehalfspacing` (interlinea 1,5);
- `\doublespacing` (interlinea 2).

Si può modificare l'interlinea soltanto in alcune parti del documento con gli ambienti `singlespace`, `onehalfspace` e `doublespace`, da usare nel modo consueto.

Se, infine, ne servisse una ancora diversa, si può dare *nel preambolo* il comando standard

```
\linespread{\fattore di scala}
```

che moltiplica l'interlinea per il *\fattore di scala*. In alternativa, il pacchetto definisce l'ambiente `spacing` da impostare come `\linespread`:

```
\begin{spacing}{\fattore di scala}
...
\end{spacing}
```

Riempimento della pagina

In mancanza di istruzioni specifiche \LaTeX cerca sempre di riempire la gabbia del testo per tutta la sua altezza. Se non riesce a farlo perché non c'è abbastanza testo, "stiracchia" il materiale che ha a disposizione inserendo

dello spazio aggiuntivo dove può: tra i capoversi, tra le voci degli elenchi e così via. Si può disattivare questo comportamento ottenendo dello spazio bianco in fondo alla pagina (scelta sconsigliata) scrivendo nel preambolo `\raggedbottom`.

3.7 STRUTTURARE IL DOCUMENTO

La tabella 9 nella pagina seguente raccoglie le principali istruzioni che producono una sezione nel documento e ne descrive il comportamento nelle classi standard. Non si considera la classe `letter` poiché non prevede alcun tipo di sezionamento.

3.7.1 Sezionare il corpo del testo e modificarne la numerazione

Per suddividere un documento in sezioni basta dare *nel corpo del testo* i comandi elencati nella prima parte della tabella, a proposito dei quali si noti quanto segue.

- I nomi inglesi dei comandi corrispondono alle unità di suddivisione del testo in vigore nei Paesi anglosassoni, e alcuni di essi non hanno corrispondenti italiani.
- Il comando `\section` produce una sezione equivalente al nostro *paragrafo*, e `\paragraph` (“capoverso”, in inglese) una sezione non numerata non equivalente né al nostro *paragrafo* né al nostro *capoverso* (che ammette al massimo un titoletto o una breve indicazione). Le stesse considerazioni valgono anche per i rimanenti comandi.
- Il comando `\part` non influenza la numerazione dei capitoli.
- Si consiglia di evitare suddivisioni così fini come quelle permesse dagli ultimi tre comandi e di assicurarsi che ogni sezione contenga *almeno due* sezioni di livello immediatamente inferiore: in caso contrario, l’unico comando di sezionamento presente diventa superfluo.

Nel documento saranno numerate automaticamente le sezioni fino a quelle di livello 2 compreso (si veda la tabella 12 a pagina 39). Per modificare la numerazione predefinita, si veda il paragrafo 3.9.3 a pagina 38.

3.7.2 Altri sezionamenti

Le altre istruzioni mostrate nella tabella 9 nella pagina successiva producono le corrispondenti sezioni descritte nei prossimi capitoli. Si noti che l’ambiente `abstract`, destinato a ospitare il sommario (o riassunto) del lavoro, è ammesso solo nelle classi `article` e `report` perché di solito nei libri è sostituito dall’introduzione.

3.7.3 Materiale iniziale, principale e finale

Oltre ai comandi appena illustrati, la sola classe `book` prevede tre dichiarazioni che agiscono al più alto livello possibile e costituiscono una specie di “supersezionamento”. Vanno date *sempre* nel corpo del documento e si comportano come segue:

- `\frontmatter` (“materiale iniziale”) non numera le sezioni e numera le pagine con numeri romani minuscoli (i, ii, iii, eccetera);

Tabella 9. Istruzioni considerate in questa guida che producono una sezione nel documento e loro comportamento nelle classi standard. I simboli indicano che la sezione possiede la caratteristica sempre (●), mai (○), solo nella classe article (⦿), solo nelle classi article e report (⦿). Tra parentesi i pacchetti richiesti.

A cosa serve	Istruzione	La sezione prodotta ha					Sezione	
		numero	titolo	testatine	posto nell'indice			
Corpo del testo	<code>\part</code>	●	●	○	●	●	Parte	
	<code>\chapter</code>	●	●	●	●	●	Capitolo	
	<code>\section</code>	●	●	●	●	●	Paragrafo	
	<code>\subsection</code>	●	●	○	●	●	Sottoparagrafo	
	<code>\subsubsection</code>	⦿	●	○	○	⦿	Sotto-sottoparagrafo	
	<code>\paragraph</code>	○	●	○	○	○	Sezione di livello ancora più basso	
	<code>\subparagraph</code>	○	●	○	○	○	Sezione al più basso livello possibile	
	Indici	<code>\tableofcontents</code>	○	●	●	○	○	Indice generale
		<code>\listoffigures</code>	○	●	●	○	○	Elenco delle figure
		<code>\listoftables</code>	○	●	●	○	○	Elenco delle tabelle
<code>\lstlistoflistings</code>		○	●	●	○	○	Elenco dei codici (listings)	
<code>\listof{...}</code>		○	●	●	○	○	Elenco di nuovi oggetti mobili (float)	
<code>\printindex</code>		○	●	●	○	○	Indice analitico (makeidx)	
Bibliografia		<code>\thebibliography</code>	○	●	●	○	○	Bibliografia manuale
	<code>\printbibliography</code>	○	●	●	○	○	Bibliografia automatica (biblatex)	
Varie	<code>abstract</code>	○	⦿	○	○	○	Sommario	

Tabella 10: Struttura generale di un libro o una tesi. Le voci in corsivo sono obbligatorie, quelle in tondo sono facoltative, quelle asteriscate *non* devono comparire nell'indice generale.

Supersezionamento	Sezione
Materiale iniziale	<i>Frontespizio</i> Colophon* Dedica* Sommario* <i>Indice generale*</i> Elenco delle figure* Elenco delle tabelle* Altri elenchi* Prefazione Ringraziamenti* Introduzione non numerata
Materiale principale	Introduzione numerata <i>Capitoli</i> Una o più appendici numerate
Materiale finale	Una o più appendici non numerate Glossario <i>Bibliografia</i> Indice analitico

- `\mainmatter` (“materiale principale”) numera le sezioni e le pagine con numeri arabi (la numerazione della pagine riprende da 1);
- `\backmatter` (“materiale finale”) non numera le sezioni e continua la numerazione araba delle pagine dal materiale principale.

3.7.4 Appendici

Per produrre le appendici basta dare la dichiarazione `\appendix`, che cambia i numeri dei capitoli (o dei paragrafi, se la classe impostata è `article`) in lettere. Il pacchetto `appendix` permette eventualmente di personalizzarle (se ne veda la documentazione).

3.7.5 Struttura generale di un libro o una tesi

La tabella 10 mostra una *possibile* successione dei componenti di una pubblicazione di una certa consistenza, come un libro o una tesi di laurea o dottorato. Lo schema è tratto con qualche variazione da [Mori, 2007], cui si rimanda per gli approfondimenti.

Si noti che:

- l'introduzione va senz'altro nel materiale principale se è a propria volta divisa in sezioni, mentre va in quello iniziale se è breve e contiene solo una sintetica esposizione dell'argomento;
- le appendici vanno valutate caso per caso in base a numero e importanza: se è una sola e poco importante può andare nel materiale finale; se invece è funzionale al corpo principale del documento va in quello principale.

Si ricordi, infine, che una sezione concepita come un capitolo non va *mai* messa nel materiale finale.

3.8 STILI DI PAGINA

Lo *stile di pagina* è l'organizzazione del contenuto di testatina e piede scelta per il documento, e va indicato nell'argomento `<stile>` del comando

```
\pagestyle{<stile>}
```

L^AT_EX prevede tre stili di pagina predefiniti e uno personalizzabile, descritti di seguito.

plain Mette i numeri di pagina nel piede, lasciando vuota la testatina. È lo stile predefinito nelle classi `article` e `report`.

empty Lascia testatina e piede vuoti.

headings Lascia il piede vuoto e compone le testatine come segue: il numero di pagina è sempre posto nel margine esterno, seguito dal titolo del capitolo corrente nella testatina di sinistra e preceduto dal titolo del paragrafo corrente in quella di destra. È lo stile predefinito nella classe `book` e agisce nello stesso modo nelle classi `report` e `article` impostate con `twoside`, con la differenza che nella seconda le testatine riportano i titoli di paragrafo e sottoparagrafo correnti, rispettivamente. Se invece s'imposta la classe con `oneside`, la testatina riporta soltanto il titolo della suddivisione maggiore.

myheadings È simile a `headings` nel risultato e va usato quando non si vuole che le testatine dipendano dai titoli delle sezioni (capitolo e paragrafo) correnti. L'utente deve specificarne il contenuto a ogni nuovo capitolo (o paragrafo, se la classe è `article`), dando `\markboth` per comporre entrambe oppure `\markright` per comporre soltanto quella di destra.

Si può cambiare lo stile della pagina *corrente* con il comando

```
\thispagestyle{<stile>}
```

Gestire le testatine

Nelle classi standard le testatine vengono prodotte dai comandi mostrati nella tabella 9 a pagina 34. Presenza o meno sulla pagina, contenuto e stile si possono regolare facilmente, ma *si raccomanda di non abusare* di queste possibilità e di attenersi alle scelte tipografiche della classe in uso. Si ricorda che le dichiarazioni di sezionamento descritte nel paragrafo 3.7.3 a pagina 33 *non* influenzano il comportamento delle testatine, che saranno dunque sempre presenti anche nelle sezioni non numerate prodotte da `\frontmatter`.

Per eliminare le testatine da una sezione che *l'utente* non ha numerato, invece, basta impostare per quella sezione lo stile di pagina `plain` (si veda il paragrafo 3.8) e ripristinare poi lo stile generale come segue:

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

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

Si noti bene che se ci si dimentica di farlo, la sezione non numerata porterà le testatine prodotte dall'ultimo comando "utile" in questo senso.

Se comunque le si volessero, vanno inserite a mano con `\markboth`

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

che si usa come segue:

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

dove `\MakeUppercase` produce il proprio argomento in tutte maiuscola (come nelle testatine standard della classe book).

Eliminare testatine inutili e aggiungere pagine bianche

Per eliminare testatine e piedi *comunque* presenti nelle pagine bianche alla fine di un capitolo usando le classi standard con l'opzione `openright` (scelta consigliata), basta caricare il pacchetto `emptypage`.

Per aggiungere una pagina bianca dopo la pagina corrente, basta caricare il pacchetto `afterpage` e dare

```
\afterpage{\null\thispagestyle{empty}\clearpage}
```

3.9 INDICE GENERALE, TITOLI E PROFONDITÀ

3.9.1 Indice generale, miniindici e indici abbreviati

Il comando

```
\tableofcontents
```

produce *nel punto in cui viene dato* la sezione contenente l'indice generale con relativi titolo e testatina. Si noti che per ottenerlo nel documento finito servono *due* composizioni successive.

In particolari circostanze potrebbe essere utile inserire anche un *miniindice* fra titolo del capitolo e inizio del testo: il pacchetto `minitoc` permette di farlo automaticamente (se ne veda la documentazione), ma non se ne abusi.

Il pacchetto `shorttoc`, invece, genera indici abbreviati della profondità scelta (come un *Sommario* o un *Indice degli argomenti*), utili in documenti particolarmente corposi per descriverne il contenuto senza scendere nei dettagli.

3.9.2 Gestire i titoli

Titoli non numerati né indicizzati

Di tutti i comandi di sezionamento elencati nella tabella 9 a pagina 34 esiste anche una *variante asterisco* formata dal nome del comando con un asterisco alla fine, che genera titoli *non numerati* e che nemmeno andranno a finire nell'indice generale, di qualunque livello essi siano. Il titolo precedente, per esempio, si è ottenuto con il comando

```
\subsubsection*{Titoli non numerati né indicizzati}
```

Per mandare nell'indice anche i titoli che normalmente non ci finirebbero, *subito dopo* il relativo comando di sezionamento basta dare

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

Tabella 11: Elementi considerati in questa guida che ammettono un riferimento incrociato.

<code>\part</code>	<code>\subparagraph</code> [*]	<code>multline</code> ^{**}	<code>listings</code> [†]
<code>\chapter</code>	<code>\footnote</code>	<code>split</code> ^{**}	<code>sidecap</code> [†]
<code>\section</code>	<code>\newtheorem</code>	<code>gather</code> ^{**}	<code>subfig</code> [†]
<code>\subsection</code>	<code>figure</code>	<code>align</code> ^{**}	<code>wrapfloat</code> [†]
<code>\subsubsection</code>	<code>table</code>	<code>enumerate</code> [‡]	n. di pagina
<code>\paragraph</code> [*]	<code>equation</code>	<code>float</code> [†]	

^{*} Solo se numerato.

^{**} Richiede il pacchetto `amsmath`.

[‡] Si richiama un elemento della lista.

[†] Rappresenta l'ambiente o gli ambienti definiti.

dove:

- *<indice>* è il tipo di indice in cui far comparire la voce in questione (normalmente si sceglie `toc`, `lot` o `lof` per l'indice generale, delle tabelle e delle figure rispettivamente);
- *<livello>* è il nome del livello di sezionamento in questione (si noti che `paragraph` e `subparagraph` non sono ammessi);
- *<titolo>* è il titolo di sezione che finirà nell'indice.

Applicato al titolo di questa sezione, il codice sarebbe:

```
\subsection*{Titoli non numerati né indicizzati}
%\phantomsection
\addcontentsline{toc}{subsection}{Titoli non numerati né indicizzati}
```

Si noti che se `hyperref` è caricato, *subito prima* di `\addcontentsline` va dato anche `\phantomsection` per evitare possibili errori nei collegamenti ipertestuali e nei segnalibri del documento finito (in tal caso, si decommenti la riga corrispondente).

Titoli alternativi nell'indice generale

Nell'indice generale finiscono i titoli scritti nell'argomento dei comandi di sezionamento. Se, però, un titolo è troppo lungo per starci agevolmente (si noti che un titolo non dovrebbe *mai* andare a capo) o si hanno particolari esigenze, lo si può sostituire con un titolo alternativo più breve, da inserire nell'argomento facoltativo degli stessi comandi:

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

Si noti che il titolo breve comparirà anche nelle testatine, se previste dalla classe di documento in uso e che, ovviamente, non si può usare se il comando è asteriscato.

3.9.3 Regolare la profondità dell'indice generale

La numerazione delle sezioni e ciò che compare nell'indice generale sono regolati dai valori numerici dei due contatori `secnumdepth` e `tocdepth` rispettivamente, entrambi impostati a 2 di default: le sezioni saranno numerate se di livello minore o uguale al valore di `secnumdepth`, mentre i loro titoli andranno nell'indice se di livello minore o uguale al valore di `tocdepth` (si veda la tabella 12 nella pagina successiva).

Tabella 12: Corrispondenza fra livelli e sezioni.

Livello	Sezione
-1	<code>\part</code>
0	<code>\chapter</code>
1	<code>\section</code>
2	<code>\subsection</code>
3	<code>\subsubsection</code>
4	<code>\paragraph</code>
5	<code>\subparagraph</code>

Per esigenze particolari (come le richieste di relatore o editore, per esempio) si potrebbe dover modificare questo comportamento predefinito. L'indipendenza dei due contatori permette di farlo facilmente, ottenendo praticamente qualsiasi risultato: si potranno avere documenti con numerazioni dettagliate e indici più snelli e viceversa.

Per esempio, scrivendo *nel preambolo*

```
\setcounter{secnumdepth}{3}
\setcounter{tocdepth}{1}
```

nel documento saranno numerate le sezioni fino al terzo livello, ma nell'indice generale compariranno solo quelle fino al primo. Come sempre, però, si raccomanda di non abusare di questa possibilità e di limitarla ai casi davvero necessari. In linea generale si sconsiglia vivamente di impostare *secnumdepth* a più di 3 e *tocnumdepth* a più di 2 (l'indice assomiglierebbe più a un elenco telefonico che a un indice vero e proprio).

3.10 RIFERIMENTI INCROCIATI

Nei documenti si trovano spesso riferimenti incrociati a sezioni, figure, tabelle, teoremi e altri elementi. Per realizzarli si usano i comandi standard

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

dove:

- `\label` assegna agli elementi mostrati nella tabella 11 a fronte un' *etichetta* arbitraria e *univoca*, che si consiglia di scrivere nella forma *abbreviazione*: *parola chiave* (dove la prima è un'abbreviazione dell'elemento in questione, come *tab* per una tabella, e la seconda una stringa identificativa) e avendo cura di evitare i caratteri accentati (scrivendo in francese, inoltre, si sconsiglia di mettere i due punti nelle etichette);
- `\ref` produce il numero dell'elemento messo in *etichetta*;
- `\pageref` produce il numero di pagina in cui l'elemento compare.

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

Per esempio, se s'identifica questo paragrafo con

```
\section{Riferimenti incrociati}
\label{sec:rif-inc}
```

poi ci si può riferire a esso con

```
Ecco un riferimento a questo
paragrafo: “si veda il
paragrafo-\ref{sec:rif-inc}”.
```

Ecco un riferimento a questo paragrafo: “si veda il paragrafo 3.10”.

Si noti che è una buona abitudine unire il riferimento alla parola precedente con uno *spazio indivisibile*: garantisce in genere risultati tipografici ottimali.

Molto spesso, specialmente quando il riferimento si trova a una o più pagine di distanza dall’oggetto, è utile avere un’indicazione completa *anche* del numero di pagina. Lo si può ottenere con il comando `\vref` del pacchetto `varioref` dopo aver messo la lingua principale del documento come opzione al comando di chiamata:

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

A questo punto `\vref` si usa come di consueto:

```
Si veda il
paragrafo-\vref{sec:pacchetti}.
```

Si veda il paragrafo 3.12 nella pagina successiva.

Se il documento contiene molti riferimenti incrociati potrebbe essere utile controllare la correttezza delle etichette o averle sempre sotto controllo: il pacchetto `showkeys` le visualizza nel margine della pagina.

3.11 COLLEGAMENTI IPERTESTUALI E AL WEB

Collegamenti ipertestuali e al Web

Il pacchetto `hyperref`, che di regola va caricato *per ultimo*, crea i collegamenti ipertestuali all’interno del documento, rendendo cliccabili i riferimenti incrociati visti nella sezione precedente, quelli a voci bibliografiche, a indirizzi Internet e molto altro. Se ne possono specificare le opzioni come di consueto oppure, se numerose, si può usare il comando `\hypersetup`:

```
\usepackage{hyperref}
\hypersetup{(chiave)=(valore),{...}}
```

Il pacchetto permette di gestire collegamenti e segnalibri in modo molto fine: se ne veda la documentazione.

Per impostazione predefinita, `hyperref` circonda il collegamento con un riquadro colorato che *non* viene stampato. Si può avere il testo del collegamento colorato (con colori predefiniti o a piacere) scrivendo:

```
\usepackage[colorlinks]{hyperref}
```

Questo è utile per un documento da leggere a schermo o da stampare a colori, ma si ricordi che la stampa in bianco e nero restituisce i colori come sfumature di grigio, a volte poco leggibili.

Si possono avere tutti i collegamenti in nero e senza riquadri scrivendo semplicemente

```
\hypersetup{hidelinks}
```

Per gestire ancora più finemente i segnalibri, che con l’opzione `bookmarks` `hyperref` gestisce automaticamente, compresi quelli delle sezioni aggiunte all’indice generale con `\addcontentsline`, si segnala il pacchetto `bookmarks`, da caricare *dopo* `hyperref` (se ne veda la documentazione).

Oltre ai collegamenti ipertestuali per i riferimenti incrociati, `hyperref` permette anche di realizzare collegamenti al Web con il comando `\href`:

```
\href{<indirizzo Internet>}{<testo del collegamento>}
```

Scrivendo

```
Visita il sito del
\href{http://www.guitex.org/}
{\GuIT*}.
```

Visita il sito del `\GuIT*`.

basta cliccare sul logo `\GuIT*` per accedere al sito omonimo.

Indirizzi Internet e di posta elettronica

Il pacchetto `url` (caricato automaticamente da `hyperref`) definisce il comando `\url`, utile per scrivere un indirizzo Internet:

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

Per i collegamenti a un indirizzo di posta elettronica conviene definire nel preambolo un apposito comando `\mail` (si veda il paragrafo 11.1 a pagina 173),

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

da usare come segue:

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

3.12 PACCHETTI

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

3.12.1 Caratteristiche

Che cosa sono?

Fondamentalmente, un pacchetto è un file “di stile” (con estensione `.sty`) scritto in linguaggio \LaTeX , contenente istruzioni che permettono di svolgere alcune operazioni.

Come sapere se servono?

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

\TeX Live non comprende *tutti* i pacchetti presenti su CTAN. Infatti, componendo un sorgente può capitare che \LaTeX produca un messaggio di errore del tipo

```
Can't find file steroid.sty
```

Ciò significa che è stato caricato un pacchetto (steroid, nell'esempio considerato) non presente nella distribuzione. Si risolve il problema seguendo le istruzioni contenute nel paragrafo 2.1.2 a pagina 14.

Viceversa, può accadere di usare un comando definito da un pacchetto che ci si è dimenticati di caricare: si otterrà un messaggio di errore di comando sconosciuto, che purtroppo non aiuta a indovinare il pacchetto che serve.

E se serve un pacchetto che non c'è in T_EX Live?

Per usare un pacchetto che *non può* esserci nella distribuzione (perché non è prevista l'inclusione in T_EX Live, o è un pacchetto personale o coperto da una licenza particolare, o è una versione sperimentale che si vuole comunque provare, o ancora perché l'aggiornamento o la pubblicazione cadono nel periodo di "congelamento") la strada più semplice è copiarne i file nella cartella di lavoro. (Ogni anno, qualche mese prima del rilascio della nuova versione gli aggiornamenti vengono "congelati". Gli archivi in Rete, tuttavia, continuano a funzionare come al solito.)

Come scovare il pacchetto che fa al proprio caso?

Questo è l'unico aspetto del lavoro con L^AT_EX in cui gusto, abilità e fortuna la fanno da padroni: cercando su CTAN (www.ctan.org) o su Sarovar (www.sarovar.org) si trovano preziosi riferimenti e soluzioni per risolvere moltissimi problemi.

3.12.2 Caricamento e precauzioni

Come caricarli?

I pacchetti si caricano *nel preambolo* con il comando

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

dove:

- *opzioni* è una voce o un elenco di voci separate con la virgola costituite da un solo elemento o un'espressione del tipo *chiave*=*valore* che specificano le impostazioni del pacchetto;
- *pacchetto* è il nome del pacchetto, che va scritto *sempre* in tutte minuscole (LayAureo si scrive così, ma si carica come *layaureo*, per intenderci).

Si noti che con uno stesso comando `\usepackage` si possono caricare più pacchetti *senza opzioni*, separandone i nomi con la virgola.

Quali precauzioni prendere?

Non si possono caricare i pacchetti in un ordine casuale, anche se ciò è permesso entro certi limiti. Sequenze di caricamento ben precise, come si è già visto per `fontenc`, `inputenc` e `babel` sono richieste anche per altri pacchetti, ma non è questa la sede per elencarle tutte (né lo si potrebbe fare per evidenti limiti di spazio). I messaggi d'errore notificati dal programma in tal senso sono chiari, di solito, ma le precauzioni da prendere non sono mai troppe. Di seguito si danno alcuni consigli che dovrebbero limitare i problemi.

- L^AT_EX richiede di caricare (direttamente o indirettamente) i pacchetti *solo nel preambolo e una volta sola*, con *tutte* le opzioni che servono.

- Molti pacchetti ne caricano automaticamente degli altri: lo si può scoprire leggendo la documentazione. Non sapendolo e ricaricando un pacchetto, si ottiene un errore.
- Talvolta il caricamento dei pacchetti sottostà a vincoli precisi: alcuni vanno caricati prima di altri e viceversa, pena un errore. La documentazione del pacchetto indicato nell'errore potrebbe contenere informazioni utili: di solito basta modificare l'ordine di caricamento o eliminare la doppia chiamata.

3.12.3 Usarli al meglio: la documentazione

Chi scrive o aggiorna un pacchetto per \LaTeX ne scrive anche, e quasi sempre in inglese, la *documentazione*, che spesso si compone di due parti distinte:

- il manuale d'uso, che dichiara lo scopo del pacchetto e ne descrive i comandi;
- il codice che costituisce il pacchetto, destinato a chi voglia eventualmente svilupparlo (nel caso di pacchetti molto corposi, il codice costituisce un file a sé).

I pacchetti contenuti in ogni distribuzione di \LaTeX sono già corredati della relativa documentazione (quasi sempre un PDF omonimo), facilmente raggiungibile con il programma `texdoc`, integrato in \TeX Live. Il programma si lancia dalla riga di comando o con le scorciatoie che di solito ogni editor definisce a questo scopo e prevede numerose opzioni di ricerca: eseguendo

```
texdoc <nome del pacchetto>
```

da una posizione qualunque sul proprio computer, in un attimo si apre il relativo manuale. Si segnala anche la versione online di `texdoc` ( `TEXDOC`), nella quale i pacchetti sono ordinati in categorie per una più agevole ricerca delle informazioni.

3.12.4 E i file `.ins` e `.dtx`?

In alcuni casi, i pacchetti che si scaricano da Internet (in forma di archivi compressi) non contengono file `.sty` e PDF di documentazione, ma un file con estensione `.ins` e uno con estensione `.dtx`. Basta procedere così:

- Il file `.dtx` contiene la documentazione del pacchetto. Eseguendo \LaTeX su di esso, si ottiene il manuale.
- Eseguendo \LaTeX sul file `.ins`, invece, si ottiene il file `.sty` (o anche più d'uno, a seconda dei casi).

Si noti che alcuni pacchetti contengono soltanto il file `.dtx`: in questo caso con la prima istruzione si ottiene *anche* il file `.sty`.

A questo punto, per poterli usare basta sistemare i file `.sty` ottenuti come descritto nel paragrafo 3.12.1 a pagina 41.

3.13 UNITÀ DI MISURA TIPOGRAFICHE

Nei prossimi capitoli spesso si useranno istruzioni che richiedono di esprimere una lunghezza in una qualche *unità di misura tipografica*. Dal momento

Tabella 13: Principali unità di misura tipografiche riconosciute da \LaTeX .

Unità	Codice	Valore
centimetro	cm	
millimetro	mm	
punto tipografico	pt	0,3514 mm
<i>x-height</i>	ex	Altezza della <i>x</i> nel font in uso
<i>m-width</i>	em	Larghezza della <i>M</i> nel font in uso

che alcune di esse poco hanno a che fare con quelle più conosciute del sistema metrico decimale, nella tabella 13 si mostrano quelle effettivamente usate in questa guida.

Esistono inoltre i comandi di spaziatura `\quad` (o *quadrato*) e `\qquad` (o *quadratone*), che producono rispettivamente uno spazio di 1 e 2 em. Si noti che in un ambiente puramente testuale il loro uso è di regola fortemente sconsigliato e va limitato a casi particolarissimi (per spaziare sottooggetti mobili, per esempio, come si spiega nel paragrafo 6.5 a pagina 116).

Per maggiori dettagli si veda [Beccari, 2012].

3.14 DOCUMENTI DI GRANDI DIMENSIONI

Per scrivere senza sorprese un documento di grandi dimensioni come un libro o una tesi è importantissimo organizzarne razionalmente il materiale. Prendendo a esempio questa guida (ma i suggerimenti valgono anche per una tesi di laurea o un altro documento), si sono messi *tutti* i suoi file in una cartella `artelatex`, strutturata in sottocartelle come segue:

- La sottocartella `inizio`, con il materiale iniziale suddiviso nei corrispondenti file come `ringraziamenti.tex`, `introduzione.tex`, eccetera.
- La sottocartella `capitoli`, con il materiale principale suddiviso nei corrispondenti file come `basi.tex`, `testo.tex`, `tabellefigure.tex`, eccetera.
- La sottocartella `fine`, con il materiale finale suddiviso nei corrispondenti file come `acronimi.tex`, `sitiinternet.tex`, eccetera.
- La cartella `immagini`, con tutte le immagini incluse nella guida. Se sono molte, le si potrebbe distribuire in ulteriori sottocartelle corrispondenti ai diversi capitoli. Immaginando di chiamarle `grafici` e `foto`, basta scrivere nel file di impostazioni (si veda poco più sotto)

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

La cartella `artelatex` *deve* contenere anche altri due file:

- il file *principale* del documento, `artelatex.tex`, cioè quello che contiene dichiarazione di classe, preambolo, impostazioni generali e ambiente `document`;
- il file `arte-bibliografia.bib` (a meno di non volerlo sistemare come spiegato nel paragrafo 7.2 a pagina 123), cioè il database bibliografico del documento.

Se il documento non è troppo corposo, invece, se ne possono mettere tutti i file in una sola cartella e l'indicazione del percorso non serve più.

Infine, si sono raccolte definizioni di comandi e ambienti personali e impostazioni generali del documento in un pacchetto `impostazioni-arte.sty` (si scrive con l'editor in uso, si registra con estensione `.sty`, *non* richiede il preambolo e *non* va composto), caricato nel preambolo come un normale pacchetto immediatamente prima dell'inizio del documento. Questi piccoli accorgimenti "puliscono" il file principale semplificando notevolmente il proprio lavoro.

Nell'ambiente `document` vanno caricati i file `.tex` in cui si è suddiviso il documento con uno dei due metodi spiegati di seguito.

3.14.1 Il metodo di `\input`

Il primo metodo prevede di scrivere il nome del file *senza l'estensione* nell'argomento del comando `\input`, indicandone l'eventuale percorso, come nell'esempio seguente:

```
\begin{document}

...
\input{inizio/ringraziamenti}
\input{inizio/introduzione}
...
\input{capitoli/basi}
\input{capitoli/installare}
\input{capitoli/testo}
...
\input{fine/acronimi}
...

\end{document}
```

Questo accorgimento evidenzia molto chiaramente la struttura del documento e snellisce il file principale. In pratica, `\input` costruisce il documento "attaccando" semplicemente uno dopo l'altro i vari file. Si noti che:

- Questi ultimi *non* devono contenere alcun preambolo, ma solo comando di sezionamento e contenuto della sezione.
- `\input` permette l'annidamento (cioè ammette altri `\input` nel proprio argomento).
- Si abbia cura che l'eventuale percorso dei file inclusi (anche quelli con `\graphicspath`) sia *relativo* (un percorso assoluto verrebbe addirittura *rifiutato* da L^AT_EX per motivi di sicurezza) e *non* contenga spazi.
- Per evitare problemi che poi sarebbe difficile risolvere, si raccomanda di nominare i file con *una sola* parola alfanumerica senza maiuscole, punti, spazi intermedi e caratteri particolari. Se fosse davvero necessario separare i due membri del nome del file, a parte `prima.tex` o `parte.prima.tex`, per esempio, si preferisca `parte-prima.tex`.

Il metodo appena spiegato è particolarmente utile per includere nel documento elementi come tabelle o grafici particolarmente complessi e possibile fonte di errori difficilmente individuabili se composti direttamente nel sorgente. Lo si può usare con profitto anche per comporre una sezione del documento alla volta (commentando quelle che non servono) con notevole risparmio di tempo. In quest'ultimo caso, però, si noti, che i contatori di pagina e di sezione vengono aggiornati a ogni composizione, e che gli eventuali riferimenti incrociati vanno perduti (in pratica, qualunque sezione si componga comincerà sempre a pagina 1 e i riferimenti incrociati alle altre verranno visualizzati con ??).

3.14.2 Il metodo di `\include` e `\includeonly`

Risolve il problema la coppia di comandi `\include` e `\includeonly`, da usare così:

```
% nel preambolo
\includeonly{introduzione,%
             basi,%
             acronimi%
            }

\begin{document}

...
\include{ringraziamenti}
\include{introduzione}
\include{basi}
...
\include{acronimi}
...

\end{document}
```

A questo punto basterà comporre prima l'intero documento e poi solo il o i file che interessano (commentando quelli che non servono nell'argomento di `\includeonly`): numerazione delle pagine e riferimenti incrociati saranno a posto. I file da includere con `\include` devono avere le stesse caratteristiche di quelli appena visti per `\input`.

Si noti che `\include`:

- ordina a \LaTeX di cominciare a comporre il contenuto del file incluso sempre su una pagina nuova: perciò è più indicato per comporre interi capitoli e non è completamente intercambiabile con `\input`;
- non permette l'annidamento, ma nulla vieta di usare `\input` nel file presente nel suo argomento;
- prima di cominciare a comporre il nuovo file, ordina a \LaTeX di svuotare le eventuali code di oggetti mobili ancora in memoria.

Come si vede, questo secondo metodo è particolarmente utile per comporre singoli capitoli di una tesi da presentare in bozza al relatore.

4 | TESTO

4.1 STRUTTURA DEL TESTO

Lo scopo principale di chi scrive un testo è comunicare idee e conoscenze al lettore, che le comprenderà tanto più quanto meglio sono strutturate, e ne apprezzerà tanto più la struttura quanto più la forma tipografica del documento rispecchia la costruzione logica del suo contenuto [Oetiker *et al.*, 2011].

La struttura interna di un'opera non letteraria di grandi dimensioni presenta quasi sempre due elementi costanti:

- la suddivisione in “blocchi” di testo di lunghezza variabile, con i blocchi più grandi che contengono quelli più piccoli;
- la presenza di altre parti aggiunte (come introduzione ed eventuali appendici).

Le suddivisioni elencate nella tabella 14 sono fondamentali per comprendere l'articolazione di un testo scritto, e vengono chiamate in generale *sezioni*.

A questo proposito si noti che:

- il sezionamento dei documenti è compito dell'utente, perché \LaTeX non lo fa automaticamente;
- è estremamente importante scandire il testo in capoversi, per chi scrive e per chi legge: le informazioni sono meglio articolate e più facilmente memorizzabili.

4.2 COMPORRE I CAPOVERSI

Spesso si sottovaluta l'importanza di scrivere un testo ben strutturato, e usando \LaTeX altrettanto frequentemente si comincia un nuovo capoverso senza nemmeno rendersene conto.

È molto facile commettere quest'ultimo errore se il testo contiene formule matematiche [Oetiker *et al.*, 2011]. Infatti l'abitudine, diffusa, di lasciare una riga vuota tra la fine di una formula e la prosecuzione del testo si sconta nel documento finito con altrettanti nuovi capoversi, anche laddove il flusso del discorso non li richiederebbe affatto.

Cominciare un nuovo capoverso

Con l'ovvia eccezione del primo capoverso di una sezione, per cominciare un nuovo capoverso con \LaTeX si hanno due possibilità:

- si lascia una riga vuota nel sorgente (di solito si fa così);
- si dà il comando `\par`.

In ogni caso, non lo si faccia *mai* con `\\` (qualche esempio di questa guida lo ha richiesto per motivi di spazio).

Tabella 14: Lunghezza orientativa delle sezioni di un testo scritto.

Sezione	Lunghezza orientativa
Parte	Imprecisabile
Capitolo	Da una decina a un centinaio di pagine
Paragrafo	Da mezza a una decina di pagine
Sottoparagrafo	Da poche righe a un paio di pagine
Capoverso	Da una a una ventina di righe
Enunciato	Da una parola a una decina di righe

Osservando gli esempi che seguono, si cerchi di capire perché a volte c'è la riga bianca e altre no. (Se non si comprendono ancora tutti i comandi, si leggano interamente questo capitolo e i primi paragrafi del capitolo 5 a pagina 65, e poi si ritorni su questo punto.)

<pre>\dots quando Einstein propose l'equazione \begin{equation} E = mc^2, \end{equation} che è allo stesso tempo la più nota e la meno compresa formula della Fisica.</pre>	<p>... quando Einstein propose l'equazione</p> $E = mc^2, \quad (4.1)$ <p>che è allo stesso tempo la più nota e la meno compresa formula della Fisica.</p>
<pre>\dots che, rispetto ai precedenti, presenta alcuni vantaggi.</pre> <p>La formula</p> <pre>\begin{equation} I_D \text{\textup{D}} = I_L \text{\textup{F}} - I_L \text{\textup{R}} \end{equation} costituisce la parte centrale di un modello molto diverso di transistor. \dots</pre>	<p>... che, rispetto ai precedenti, presenta alcuni vantaggi.</p> <p>La formula</p> $I_D = I_F - I_R \quad (4.2)$ <p>costituisce la parte centrale di un modello molto diverso di transistor. ...</p>
<pre>\dots da cui segue la legge di Kirchhoff sulle correnti: \begin{equation} \sum_{k=1}^n I_k = 0. \end{equation}</pre> <p>La legge di Kirchhoff sulle tensioni può essere ricavata\dots</p>	<p>... da cui segue la legge di Kirchhoff sulle correnti:</p> $\sum_{k=1}^n I_k = 0. \quad (4.3)$ <p>La legge di Kirchhoff sulle tensioni può essere ricavata...</p>

Terminare un capoverso con una formula in display (si veda il paragrafo 5.1 a pagina 65) come nel terzo degli esempi proposti è raro, ma comunque lecito. È invece *sempre* sconsigliabile cominciarlo con una formula matematica di qualunque tipo.

Capoversi ben composti e microtipografia

Un documento “ben composto” si riconosce da alcuni elementi: il testo è giustificato, le righe sono interamente riempite, le parole sono adeguate-mente spaziate tra loro e sillabate a fine riga se proprio non ci stanno. I

capoversi presentano la prima riga rientrata per facilitare la lettura e non sono solcati dai fastidiosi “ruscelli” (le zone bianche che serpeggiano sulla pagina in verticale). Di solito tutto questo si ottiene dando a mano le rispettive impostazioni; con \LaTeX , invece, non occorre nemmeno pensarci, perché il programma:

- giustifica il testo per impostazione predefinita;
- riempie la riga con un sofisticatissimo algoritmo di spaziatura tra le parole, sillabandole solo se *davvero* necessario;
- rientra automaticamente la prima riga di ogni capoverso tranne il primo (se per qualche motivo non si volesse il rientro, basta cominciare la riga interessata con `\noindent`);
- numera automaticamente le pagine del documento;
- non aggiunge spazio supplementare tra un capoverso e l’altro tranne quando non ha abbastanza materiale per riempire perfettamente la pagina.

Talvolta, invece, questo spazio supplementare potrebbe servire. Lo si può inserire con i seguenti comandi:

- `\bigskip`, `\medskip` e `\smallskip`, avendo cura di lasciare una riga bianca *prima*, inseriscono uno spazio verticale rispettivamente “grande”, “medio” e “piccolo” la cui ampiezza è in funzione del font utilizzato.
- `\vspace{⟨lunghezza⟩}` inserisce uno spazio verticale pari a `⟨lunghezza⟩` (che va perso se dopo la composizione viene a trovarsi all’inizio di una pagina: per mantenerlo basta usare la forma `\vspace*.`)

La tipografia anglosassone (predefinita in \LaTeX) non prevede il rientro della prima riga del primo capoverso di una sezione. Per ottenerlo, secondo una consuetudine spesso seguita in Italia, basta semplicemente caricare il pacchetto `indentfirst` nel modo consueto.

Il pacchetto `microtype` migliora il riempimento delle righe abilitando:

- *l’espansione dei font*, ovvero espande i caratteri per riempire la riga in modo ancora migliore;
- *la protrusione dei caratteri*, ovvero permette a certi caratteri di sporgere leggerissimamente a fine riga (di solito segni di punteggiatura e trattini).

Perciò si consiglia di caricarlo *sempre*, anche se la prima delle due funzionalità appena descritte ancora non funziona con \XeTeX (si veda il paragrafo [B.2.2](#) a pagina [204](#)).

Interrompere una riga senza cominciare un nuovo capoverso

In casi particolari può essere necessario interrompere una riga. Per farlo si usano i comandi `\\` o `\newline`, e se ne incomincia una nuova *senza iniziare un nuovo capoverso* (e senza rientro, dunque, come qui).

Si può inserire uno spazio aggiuntivo tra due linee dello stesso capoverso con il comando `\\[⟨lunghezza⟩]`, in cui `⟨lunghezza⟩` può essere espressa in una qualunque delle unità di misura tipografiche accettate da \LaTeX , avendo cura di usare *il punto* come separatore decimale (si veda la tabella [13](#) a pagina [44](#)).

Dividere le parole a fine riga

In generale, \LaTeX cerca di interrompere le righe sempre nel miglior punto possibile. Se, però, non riesce a farlo neppure secondo i propri severi criteri, le lascia fuoriuscire dal margine destro e avverte l'utente con un messaggio di *Overfull hbox*. Non sempre è facile individuarle: nel capitolo 12 a pagina 185 si spiega come fare.

L'algoritmo di sillabazione di \LaTeX funziona correttamente con quasi tutte le parole, ma in particolari circostanze si potrebbe volere una divisione diversa da quella automatica. Con nomi propri o tecnicismi come *nitro-idrossilamminico* o *macroistruzione*, per esempio, a volte si richiede la sillabazione etimologica anziché quella che \LaTeX esegue di default: *nitro-idrossilamminico* invece di *ni-troi-dros-si-lam-mi-ni-co* e *ma-cro-i-stru-zio-ne* invece di *ma-croi-stru-zio-ne*.

In questi casi basta scrivere le parole nell'argomento di `\hyphenation` (nel preambolo) già *sillabate*, separandole con uno spazio ed evitando caratteri speciali e simboli:

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

Il comando precedente funziona anche al contrario. Una scrittura come la seguente

```
\hyphenation{nitro-idrossil-amminico FORTRAN}
```

sillaba *nitroidrossilamminico* e *Nitroidrossilamminico* come suggerito nell'argomento, ma non *FORTRAN*, *Fortran* e *fortran*. Analogamente, si può usare `\hyphenation` per forzare qualunque cesura si desideri: se si vuole spezzare la parola *melograno* soltanto tra *melo* e *grano*, si scrive:

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

Se la parola in questione compare nel documento una sola volta, si può suggerirne la sillabazione direttamente nel testo. Il comando `\-` spezza la parola nel punto (o nei punti) in cui viene dato, e *in quel punto soltanto*.

La scoperta dell'acido
nitro\idrossil\amminico
avvenne nel 1896.

La scoperta dell'acido nitroidrossil-
amminico avvenne nel 1896.

Si noti che anche gli interventi sulla sillabazione, come tutti quelli operati "a mano" sul documento, dovrebbero essere effettuati durante la revisione finale immediatamente precedente la stampa. La prima cura per un *Overfull hbox*, per esempio, dovrebbe consistere *sempre* nel riformulare l'enunciato piuttosto che nell'imporre una sillabazione particolare.

L'opzione *italian* di *babel* definisce il comando `/`, utile per andare a capo dopo la barra nelle espressioni che comportano l'alternanza tra due possibilità. La scrittura *modulazione/demodulazione*, per esempio, produce *modulazione/demodulazione* se si trova all'interno di una riga, *modulazione/demodulazione* se invece si trova alla fine.

Il comando

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

serve per mantenere unita una parola *senza* usare `\hyphenation`. Va usato all'occorrenza, magari perché in un certo punto del documento non va bene che la parola sia spezzata, ma altrove sì:

Entro quest'anno avrò imparato il
Fortran. \\[lex]
Entro quest'anno avrò imparato il
`\mbox{Fortran}`.

Entro quest'anno avrò imparato il For-
tran.
Entro quest'anno avrò imparato il
Fortran.

Spazi interparola e punti fermi

Per giustificare i capoversi \LaTeX inserisce spazi interparola variabili e migliora la leggibilità del testo separando gli enunciati con uno spazio leggermente più ampio di quello inserito da un comune elaboratore di testo. Il programma interpreta diversamente il punto:

- un punto (fermo, interrogativo o esclamativo) dopo una *minuscola* indica la fine di un enunciato, e dopo di esso \LaTeX inserisce uno spazio supplementare;
- un punto dopo una *maiuscola* indica la fine di un'abbreviazione, e dopo di esso ci sarà uno spazio normale.

Le eccezioni alle regole generali appena esposte vanno specificate esplicitamente. I casi sono tre:

- immediatamente *dopo* un punto di fine abbreviazione dentro un enunciato (tranne se l'abbreviazione ne è l'ultima parola), si usa `_`;
- immediatamente *prima* di un punto di fine enunciato che segue una maiuscola (che per \LaTeX indica *comunque* un'abbreviazione) si usa `\@`;
- per tenere unite espressioni che non si vogliono o non possono *mai* essere spezzate da un fine riga si usa lo *spazio indivisibile* prodotto dalla tilde `~`.

L'esempio seguente mostra `\@` all'opera:

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

La spaziatura corretta è quella prodotta dalla seconda scrittura.

Si osservi come agisce la tilde nei due esempi seguenti:

Avevamo letto le dispense del prof. Beccari. \[lex] Avevamo letto le dispense del prof.~Beccari.	Avevamo letto le dispense del prof. Beccari. Avevamo letto le dispense del prof. Beccari.
Il concetto è spiegato nel paragrafo \ref{sec:par}. \[lex] Il concetto è spiegato nel paragrafo~\ref{sec:par}.	Il concetto è spiegato nel paragrafo 4.2. Il concetto è spiegato nel paragrafo 4.2.

Come si può notare, la seconda scrittura di ciascuna coppia, che è quella corretta, evita che le righe finiscano o comincino nel modo sbagliato.

Per disabilitare lo spazio supplementare dopo un punto *in tutto* il documento (come nella tipografia francese) anche se si è usato `\@`, basta dare nel preambolo la dichiarazione `\frenchspacing`. Si possono comporre in questo modo soltanto alcune parti del lavoro dandola direttamente nel corpo del documento (in tal caso, non serve più `\@` ogni volta) e annullandola con `\nonfrenchspacing`.

Si noti che \LaTeX *non* mette automaticamente in maiuscolo l'iniziale della prima parola dopo un punto (di qualunque specie sia), a differenza di quanto accade con la maggior parte degli altri programmi di videoscrittura.

Tabella 15: Virgolette, tratti e puntini di sospensione. Per evidenziare le differenze tra i vari segni, virgolette alte e apici sono composti con il font Computer Modern.

	Segno	Codice	Risultato
Virgolette	semplici alte	' '	' '
	doppie alte	‘ ’	“ ”
		“ ”	“ ”
	doppie basse	<< >>	« »
« »		« »	
Tratti	trattino	-	-
	tratto	--	—
	lineetta	---	—
	meno	\$-\$	—
Puntini		\dots	...

4.3 CARATTERI PARTICOLARI E SIMBOLI

4.3.1 Virgolette, tratti e puntini di sospensione

Virgolette

In tipografia si usano comunemente tre tipi di virgolette: gli ‘apici’, le “virgolette inglesi” e le «virgolette caporali». I modi per ottenerle sono molti, e dipendono dalla codifica di input impostata, dall’editor in uso e dalle particolari sequenze di tasti digitate: la tabella 15 raccoglie quelli più usati (scopra l’utente le poche alternative).

Gli esempi seguenti le mostrano all’opera:

Ora dovrebbe essere chiaro il concetto di 'composizione asincrona'.	Ora dovrebbe essere chiaro il concetto di 'composizione asincrona'.
La Delta di Dirac è una “funzione impropria”.	La Delta di Dirac è una “funzione impropria”.
<<Se stavi attento, Ermanno, capivi tutto anche tu.>>	«Se stavi attento, Ermanno, capivi tutto anche tu.»

Si noti che nel font principale di questa guida virgolette inglesi e apici aperti e chiusi sono molto simili (ciò non accade con altri font).

Trattini, tratti e lineette

La tipografia distingue quattro tipi di tratto: tre (*trattino*, *tratto* e *lineetta*) corrispondono a un numero crescente di trattini consecutivi, mentre il quarto è il segno matematico *meno*. La tabella 15 mostra come ottenerli, e gli esempi seguenti ne illustrano alcuni possibili usi:

Stratford-on-Avon, e-mail \\ p.~13-67, 1921-28 \\ Ottica----Schema generale \\ ----Eccomi---- disse. \\ \$0\$, \$1\$ e \$-1\$	Stratford-on-Avon, e-mail p. 13-67, 1921-28 Ottica – Schema generale — Eccomi — disse. 0, 1 e -1
---	--

Tabella 16: Loghi frequenti nel mondo \LaTeX .

Comando	Risultato	Pacchetto richiesto
<code>\TeX</code>	\TeX	
<code>\LaTeX</code>	\LaTeX	
<code>\LaTeXe</code>	$\LaTeX_{2\epsilon}$	
<code>\XeTeX</code>	\XeTeX	metalogo
<code>\XeLaTeX</code>	\XeLaTeX	metalogo
<code>\AmS</code>	$\mathcal{A}\mathcal{M}\mathcal{S}$	amsmath
<code>\MF</code>	METAFONT	mflogo
<code>\MP</code>	METAPOST	mflogo
<code>\GuIT, \GuIT*</code>	$\mathcal{G}\mathcal{U}\mathcal{I}\mathcal{T}$	guit
<code>\Ars</code>	$\mathcal{A}\mathcal{R}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}\mathcal{N}\mathcal{I}\mathcal{C}\mathcal{A}$	guit
<code>\BibTeX</code>	$\text{BIB}\TeX$	dtklogos
<code>\MiKTeX</code>	$\text{MiK}\TeX$	dtklogos

Puntini di sospensione e segno di omissione

Se inseriti battendo tre punti consecutivi, i *puntini di sospensione* potrebbero compromettere la spaziatura tra le parole o la corretta interruzione di riga. \LaTeX risolve il problema definendo il comando `\dots`, che li produce correttamente spaziati e li tiene uniti *in ogni caso*:

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

Per indicare l'omissione di una parola o una porzione di testo originali, si usa il segno di *omissione* [...] definendo nel preambolo un comando ad hoc `\omissis` (si veda il paragrafo 11.1 a pagina 173):

```
\newcommand{\omissis}{[\textellipsis\unkern]}
```

Si noti che entrambi i comandi appena visti “producono testo”: prima della parola successiva si metta uno spazio esplicito nei modi consueti.

4.3.2 Loghi, accenti, caratteri particolari, apici e pedici

Loghi, accenti e caratteri particolari

La tabella 16 elenca i comandi che producono i loghi più comuni nel mondo \LaTeX (si veda la documentazione dei pacchetti dtklogos e hologo per ulteriori loghi).

\LaTeX permette di usare accenti e caratteri particolari di molte lingue (nella tabella 17 nella pagina successiva sono esemplificati per la lettera *o*, ma funzionano anche per tutte le altre lettere), come si può vedere nell'esempio seguente:

H\^otel, na"if, Stra\ss e, !'Se\~norita!, \e\l\ev e, Sm\o rrebr\o d, z\l oty	H\u00f4tel, na\u00eff, Stra\u00dfe, \u00a7Se\u00f1orita!, \e\l\ev e, Sm\u00f8rrebr\u00f8d, z\u0142oty
--	--

Il simbolo ufficiale dell'euro (€), da mettere sempre *dopo* l'eventuale numero, si ottiene con il comando `\euro` del pacchetto eurosym.

Apici e pedici

L'opzione `italian` di `babel` definisce una coppia di comandi che producono il proprio argomento in tondo *anche in modo matematico*. In modo testuale, inoltre, mantengono anche lo stile corrente, qualunque sia:

Tabella 17: Accenti e caratteri particolari.

<code>\‘o</code>	ò	<code>\u{o}</code>	ö	<code>\~o</code>	õ	<code>\.o</code>	ó
<code>\’o</code>	ó	<code>\t{oo}</code>	ôo	<code>\r{o}</code>	ô	<code>\d{o}</code>	o
<code>\^o</code>	ô	<code>\"o</code>	ö	<code>\c{o}</code>	ç	<code>\=o</code>	ô
<code>\v{o}</code>	ö	<code>\H{o}</code>	ö	<code>\k{o}</code>	ç	<code>\b{o}</code>	o
<code>\OE</code>	Œ	<code>\AE</code>	Æ	<code>\AA</code>	Å	<code>\O</code>	Ø
<code>\oe</code>	œ	<code>\ae</code>	æ	<code>\aa</code>	å	<code>\o</code>	ø
<code>\L</code>	Ł	<code>\DH</code>	Đ	<code>\DJ</code>	Đ	<code>\TH</code>	Þ
<code>\l</code>	ł	<code>\dh</code>	đ	<code>\dj</code>	đ	<code>\th</code>	þ

- `\ap{<testo>}` produce un apice come nelle abbreviazioni oggi in disuso *sig.^{ra}* o *f.^{lli}* (l’alternativa è `<testo>`);
- `\ped{<testo>}` produce un pedice, utile per qualche sostanza chimica come la vitamina B₁₂, per esempio.

Per l’elenco completo di *tutti* i simboli e i caratteri speciali di L^AT_EX (quasi 6000), si veda [Pakin, 2009].

4.4 MODIFICARE STILE E CORPO DEL FONT

4.4.1 Modificare lo stile

I comandi elencati nella tabella 18 a fronte modificano lo *stile* del proprio argomento (e solo di quello), lasciando invariato il testo successivo:

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

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

I comandi si possono combinare, ma la combinazione richiesta potrebbe non esserci nel font in uso, come mostrano gli esempi seguenti:

L’espressione che segue
`\textit{\textbf{è in nero corsivo}}`, ma questa
`\textsc{\textit{non è in maiuscoletto corsivo}}`.

L’espressione che segue è *in nero corsivo*, ma questa *non è in maiuscoletto corsivo*.

A ciascun comando corrisponde una dichiarazione che si comporta come spiegato nel paragrafo 3.4.1 a pagina 22. Anche le dichiarazioni si possono combinare:

L’espressione che segue
`{\itshape è in {\bfseries nero corsivo}}`.

L’espressione che segue è *in nero corsivo*.

ma si consiglia di usarle, se proprio necessarie, per porzioni di testo consistenti e non per singole parole come qui.

Tabella 18: Comandi e dichiarazioni per modificare lo stile del font. Per evidenziare la differenza dal corsivo, lo stile inclinato è composto con il font Computer Modern.

Comando	Dichiarazione	Stile
<code>\emph</code>	<code>\em</code>	<i>Evidenziato</i>
<code>\textit</code>	<code>\itshape</code>	<i>Corsivo</i>
<code>\textsc</code>	<code>\scshape</code>	MAUSCOLETTTO
<code>\textbf</code>	<code>\bfseries</code>	Nero
<code>\textsl</code>	<code>\slshape</code>	<i>Inclinato</i>
<code>\textrm</code>	<code>\rmfamily</code>	Tondo
<code>\textsf</code>	<code>\sffamily</code>	Senza grazie
<code>\texttt</code>	<code>\ttfamily</code>	Macchina per scrivere

4.4.2 Modificare il corpo

L'effettivo corpo del font in un documento dipende da tre fattori:

- la classe di documento scelta;
- l'opzione di corpo (eventualmente) assegnata alla classe;
- le (eventuali) dichiarazioni per modificare il corpo del font date nel testo.

Le dichiarazioni elencate nella tabella 19 nella pagina seguente modificano il *corpo* del font. Anche il contenuto della tabella risente dei fattori appena elencati: in particolare, `\normalsize` è il corpo del testo principale di questa guida.

```
Lettere {\Large grandi} e
{\scriptsize piccole}.
```

Lettere **grandi** e piccole.

Si noti che le dichiarazioni appena viste *modificano anche l'interlinea* del capoverso interessato, com'è giusto che sia, ma solo se esso termina entro il loro raggio d'azione. Nei due esempi seguenti, `\par` produce effetti differenti a seconda di dove lo si dà.

```
{\large Socrate: «Platone
mentirà nella frase
seguente»}.\par
```

Socrate: «Platone mentirà
nella frase seguente».

```
{\large Platone: «Socrate ha
detto il vero nella frase
precedente»}.\par}
```

Platone: «Socrate ha detto il
vero nella frase precedente».

Come si può osservare nel primo esempio, fuori dal gruppo `\par` non funziona più, con un risultato finale poco gradevole. La scrittura corretta pertanto è la seconda.

Questo viaggio tra stili e dimensioni si conclude con un simpatico consiglio, che starà all'utente seguire o meno:

Ricorda! *Quanti Più* corpi e stili **scegli** di usare in un documento, *tanto* più LEGGIBILE e bello *diventa*.

Tabella 19: Dichiarazioni per modificare il corpo del font.

Dichiarazione	Risultato
<code>\tiny</code>	Esempio
<code>\scriptsize</code>	Esempio
<code>\footnotesize</code>	Esempio
<code>\small</code>	Esempio
<code>\normalsize</code>	Esempio
<code>\large</code>	Esempio
<code>\Large</code>	Esempio
<code>\LARGE</code>	Esempio
<code>\huge</code>	Esempio
<code>\Huge</code>	Esempio

4.5 TITOLI E FRONTESPIZI

Titoli standard

Il comando

```
\maketitle
```

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

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

Il loro funzionamento si spiega da sé, ma si osservi quanto segue:

- un `<titolo>` troppo lungo per stare su una sola riga si spezza con `\\` (ma lo si eviti il più possibile);
- i vari `<autore>` di un documento scritto a più mani si separano con `\and`;
- \LaTeX stampa come predefinita la `<data>` della composizione anche se `\date` non viene dato, mentre la omette lasciandone vuoto l’argomento (`\date{}`).

Per inserire ringraziamenti veloci si usa il comando

```
\thanks{<ringraziamenti>}
```

che nelle classi standard e produce il proprio argomento come una nota al piede con un simbolo a esponente. Lo si può dare *dentro* l’argomento di uno qualunque dei tre comandi appena esaminati. Ecco un esempio:

```
\author{Lorenzo Pantieri \and Tommaso Gordini%
\thanks{Ringraziamo i membri del \GuIT.}}
```

Nelle classi \mathcal{AMS} `\thanks` va invece dato in una riga a sé e *fuori* dai comandi.

Frontespizio

Il titolo generato dal comando `\maketitle`, si deve riconoscere, è piuttosto spartano, anche se si può accettare in articoli e relazioni. Si consiglia di



Figura 6: Esempio d'uso di frontespizio.

comporre il frontespizio di una tesi di laurea o di dottorato con il pacchetto frontespizio (si veda la figura 6). Il pacchetto, personalizzabile, permette di inserire tutti i dati necessari, prevede opzioni per usare i diversi stili di carattere e inserire loghi universitari e immagini in filigrana. Se ne veda la ricca documentazione (in italiano).

In alternativa si può usare la *suite* ClassicThesis, che comprende un modello di tesi pronto per l'uso completo di frontespizio [Pantieri, 2007].

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

4.6 NOTE A MARGINE E A PIÈ DI PAGINA

In linea generale si usino le note *con grande moderazione*: specie quelle al piede, infatti, interrompono la lettura e possono creare seri problemi d'impaginazione. Si tenga presente che una nota *deve* potersi omettere leggendo: se il suo contenuto si rivela essenziale alla comprensione del discorso, evidentemente va levato dalla nota e messo nel corpo del testo.

Note a margine

Una nota di questo tipo si ottiene molto semplicemente con il comando

```
\marginpar{testo della nota a margine}
```

Nei documenti impostati per la stampa in fronte/retro le note vengono stampate nel margine destro nelle pagine dispari e nel margine sinistro in quelle pari. Nei documenti solo fronte, invece, saranno sempre nel margine destro. (Se una o più note dovessero comparire nel margine sbagliato, basta caricare il pacchetto `mparhack`.)

Note a piè di pagina

Il comando

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

produce una nota in fondo alla pagina corrente con un riferimento nel testo costituito da un numero a esponente. Le note al piede dovrebbero essere messe, per quanto possibile, alla fine del relativo capoverso *dopo* il punto fermo.¹

Le note a piè di pagina sono l'emblema della meticolosità.% \footnote{Eccone un esempio.}	Le note a piè di pagina sono l'emblema della meticolosità. ^a a Eccone un esempio.
---	--

Si tenga presente che:

- la loro numerazione riprende a ogni `\chapter` o `\section`;
- se sono poche o pochissime, anziché il riferimento numerico predefinito se ne consiglia uno simbolico: basta scrivere nel preambolo

```
\renewcommand{\thefootnote}{\fnsymbol{footnote}}
```

\LaTeX produce note di alta qualità, ma quando circostanze particolari le richiedono in un formato diverso da quello predefinito basta caricare il pacchetto `footmisc` e configurarlo opportunamente (se ne veda la documentazione).

4.7 EVIDENZIARE LE PAROLE

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

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

```
\emph{<testo>}
```

che si vede all'opera nell'esempio seguente:

<pre>\emph{All'interno di un testo già evidenziato, \LaTeX{ evidenzia ulteriormente con lo \emph{stile tondo}.}</pre>	<p><i>All'interno di un testo già evidenziato, \LaTeX evidenzia ulteriormente con lo stile tondo.</i></p>
---	--

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

<pre>\emph{Sarò lì in \textit{dieci} minuti.} \\ \textit{Sarò lì in \emph{dieci} minuti.}</pre>	<p><i>Sarò lì in dieci minuti.</i> <i>Sarò lì in dieci minuti.</i></p>
---	--

¹ Così.

4.8 AMBIENTI TESTUALI

4.8.1 Elenchi puntati, numerati e descrizioni

In un documento gli elenchi sono molto importanti. Infatti:

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

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

Gli elenchi:

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

Gli elenchi:

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

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

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

Ecco un elenco numerato:

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

Ecco un elenco numerato:

1. Mane;
2. Tekel;
3. Fares.

Si noti che è opportuno usare una lista numerata se in seguito ci si deve riferire a un suo elemento particolare (anche assegnandogli un’etichetta) o se per esempio si devono elencare le fasi di un procedimento. Altrimenti è più opportuna una lista puntata.

L’ambiente `description` si usa per le *descrizioni*, elenchi in cui il segno distintivo è una parola o un’espressione che si deve descrivere o spiegare, da scrivere nell’argomento facoltativo (*in questo caso*, però, obbligatorio) di `\item`:

E ora una descrizione:

```
\begin{description}
\item[itemize] Per gli
      elenchi puntati.
\item[enumerate] Per gli
      elenchi numerati.
\item[description] Per gli
      elenchi in cui ogni
      elemento comincia con
      un testo a piacere.
\end{description}
```

E ora una descrizione:

```
ITEMIZE Per gli elenchi puntati.
ENUMERATE Per gli elenchi numerati.
DESCRIPTION Per gli elenchi in cui
      ogni elemento comincia con un
      testo a piacere.
```

Come si può osservare, \LaTeX evidenzia automaticamente l’argomento del comando secondo le impostazioni generali della classe di documento in uso.

\LaTeX permette di annidare anche gli elenchi (mai più di una lista dentro l'altra, però), come mostra l'esempio seguente:

Gli elenchi:

```
\begin{itemize}
\item sono facili da usare;
\item rendono più chiaro
      il testo:
      \begin{itemize}
\item articolandolo;
\item facilitandone
      la lettura;
\end{itemize}
\item permettono di strutturare
      meglio il pensiero.
\end{itemize}
```

Gli elenchi:

- sono facili da usare;
- rendono più chiaro il testo:
 - articolandolo;
 - facilitandone la lettura;
- permettono di strutturare meglio il pensiero.

Si noti che \LaTeX cambia automaticamente il contrassegno negli elenchi annidati, che si possono individuare più facilmente nel sorgente rientrandoli leggermente (può essere una buona abitudine, ma non è una regola).

Di seguito si riportano alcune convenzioni tipografiche comunemente seguite nella composizione delle liste (le stesse osservate in questa guida):

- ogni voce di un elenco *semplice* (i cui elementi sono costituiti da un solo enunciato) comincia con l'iniziale minuscola e termina con il punto e virgola tranne l'ultima, seguita dal punto fermo;
- ogni voce di un elenco *complesso* (in cui *almeno uno* degli elementi sia composto da più di un enunciato) comincia con l'iniziale maiuscola (anche dopo il segno di due punti) e termina con il punto fermo.

Non bisogna per forza uniformare *tutti* gli elenchi di un documento a criteri stabiliti a priori: l'importante è essere coerenti volta per volta.

4.8.2 Allineare e centrare i capoversi

\LaTeX definisce tre ambienti standard per allineare un capoverso a sinistra:

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

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

a destra:

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

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

o per centrarlo sulla pagina:

```
\begin{center}
Al centro \\  
dell'universo.
\end{center}
```

Al centro
dell'universo.

Come si può osservare, il testo va a capo automaticamente, a meno di un'interruzione esplicita con `\\`.

Dedica

Non esistono regole tipografiche vincolanti per produrre la dedica in una pubblicazione, se non il gusto e le esigenze dell'utente. In linea generale, se di dimensioni contenute si colloca orizzontalmente sulla pagina con uno degli ambienti appena considerati.

Altrettanto libero ne è il collocamento in verticale, che si può controllare con i comandi opportuni: qualche prova permette di ottenere il risultato desiderato.

4.8.3 Citazioni e intercitazioni

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

Citazioni in linea

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

Una citazione in linea è un'espressione appartenente al flusso del discorso:
«il cielo stellato sopra di me,
la legge morale dentro di me».

Una citazione in linea è un'espressione appartenente al flusso del discorso:
«il cielo stellato sopra di me, la legge morale dentro di me».

Citazioni di questo tipo sono generalmente brevi.

Citazioni in display

Una citazione "in display" è un testo che va composto entro margini più ampi di quelli correnti e separandolo dal contesto con adeguati spazi bianchi, in modo da metterlo "in mostra" e bene in risalto sulla pagina.

I due ambienti standard definiti da \LaTeX allo scopo, `quote` e `quotation`, non sono del tutto soddisfacenti perché, per esempio, non riducono automaticamente il corpo del testo citato come richiedono le buone tradizioni tipografiche. Per ottenere citazioni in display ben composte si consiglia il pacchetto `quoting`, da impostare come segue nel preambolo se si prevede di mantenere lo stesso stile in tutte le citazioni del documento (scelta consigliata):

```
\usepackage{quoting}
\quotingsetup{font=small}
```

Il pacchetto definisce l'omonimo ambiente `quoting` da usare così:

Una citazione in display è un testo che \LaTeX compone su linee a sé:
`\begin{quoting}`
Il cielo stellato sopra di me,
la legge morale dentro di me.
`\end{quoting}`
Come si può osservare, la citazione è centrata e separata dal resto del testo.

Una citazione in display è un testo che \LaTeX compone su linee a sé:

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

Come si può osservare, la citazione è centrata e separata dal resto del testo.

Si noti che la citazione può essere lunga a piacere e suddivisa in capoversi, se necessario. In particolare:

- se la citazione continua il discorso principale non deve avere la prima riga rientrata: in questo caso nel sorgente `quoting` seguirà immediatamente il testo che precede *senza righe bianche in mezzo*;
- se la citazione comincia un nuovo capoverso, invece, il rientro ci va: basta lasciare una riga bianca tra l'ambiente e il testo precedente.

In entrambi i casi (da intendere come suggerimenti più che come regole universalmente accettate), la prima riga dei capoversi successivi al primo sarà rientrata per impostazione predefinita.

Per ulteriori esigenze (citazioni in lingua straniera, indicazione della fonte, eccetera) si rimanda al paragrafo [11.2.2](#) a pagina [177](#) e al pacchetto `csquotes`.

Citazioni annidate

Per le *citazioni annidate* (cioè citazioni dentro altre citazioni) le convenzioni oscillano, perciò la cosa migliore è decidere per l'una o l'altra delle seguenti possibilità e mantenerla in tutto il documento. In generale:

- se è in linea, si può mettere la citazione tra virgolette diverse da quelle scelte per la citazione principale;
- se è in `display`, la si può mettere tra virgolette oppure in un altro ambiente `quoting` annidato in quello principale.

4.8.4 Poesie

Per scrivere poesie \LaTeX definisce l'ambiente `verse`, nel quale:

- i margini vengono aumentati come per le citazioni;
- ogni verso, tranne l'ultimo della strofa, *deve* finire con `\\`;
- le strofe vengono separate automaticamente tra loro con dello spazio bianco.

L'esempio seguente mostra l'ambiente all'opera:

<hr/> <pre>La poesia <code>\emph{Un'altra notte}</code> è di Giuseppe Ungaretti. <code>\begin{verse}</code> In quest'oscuro \\ colle mani \\ gelate \\ distinguo \\ il mio viso Mi vedo \\ abbandonato nell'infinito <code>\end{verse}</code></pre> <hr/>	<pre>La poesia <i>Un'altra notte</i> è di Giuseppe Ungaretti. In quest'oscuro colle mani gelate distinguo il mio viso Mi vedo abbandonato nell'infinito</pre>
--	---

Per esigenze poetiche più avanzate è utile il pacchetto `verse` (se ne veda la documentazione).

4.8.5 Codici e algoritmi

Talvolta capita di dover scrivere parole o frammenti di testo in modo *verbatim* (“alla lettera”), cioè *non* interpretando spazi, caratteri speciali, rientri, a capo, simboli e comandi, che potrebbero avere un’importanza particolare e devono rimanere tali. Questa modalità è utile per riportare esempi di codici informatici e linguaggi di programmazione, come si è fatto in questa guida tutte le volte che si è mostrata la sintassi dei comandi e degli ambienti di \LaTeX .

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

```
\verb!<testo verbatim>!
```

Il carattere `!` è solo uno dei possibili *caratteri delimitatori*, cioè con la sola funzione di indicare inizio e fine del *<testo verbatim>*: a questo scopo si può usare un carattere qualunque, *tranne **, purché non compaia tra i caratteri da riprodurre. Se ne consiglia uno tra `! ? | @`. Si noti che \LaTeX *non* sillaba il *<testo verbatim>*: se troppo lungo, infatti, sporgerà nel margine destro.

Per scrivere testo *verbatim in display e su più righe*, invece, c’è l’ambiente standard `verbatim`, da usare come di consueto. A questo proposito si noti che:

- né `\verb` né `verbatim` possono comparire nell’argomento di un altro comando;
- `verbatim` può riprodurre tutto *tranne* `\end{verbatim}` (l’ambiente sarebbe chiuso due volte, infatti, e l’errore arriverebbe inesorabile).

Sia `\verb` sia `verbatim` prevedono una variante asterisco che riproduce lo spazio in modo visibile con il carattere `_`, come si può osservare negli esempi seguenti.

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

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

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

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

```
\begin{verbatim*}
Il comando \verb*
e l'ambiente verbatim*
mostrano gli spazi così.
\end{verbatim*}
```

Il comando `\verb*`
e l’ambiente `verbatim*`
mostrano gli spazi così.

Questi strumenti generalmente riescono a soddisfare le esigenze più comuni, ma non si possono personalizzare in alcun modo (con colori e sfondi particolari, riquadri, possibilità di definire ambienti e linguaggi personali, per esempio). Si rimanda chi ne avesse bisogno ai pacchetti `listings` (con le limitazioni di cui si parla nel capitolo 9 a pagina 139) e `fancyvrb`.

Per scrivere algoritmi si consigliano i pacchetti `algorithm` e `algpseudocode`: il primo, a differenza del secondo, genera oggetti mobili (si veda a questo proposito il paragrafo 6.2 a pagina 92).

4.9 ACRONIMI E GLOSSARI

In lavori particolarmente tecnici e complessi si consiglia di mettere nel *backmatter* un elenco degli acronimi menzionati nel testo e un *glossario*, cioè l'elenco alfabetico di una scelta di termini (di solito specialistici) presenti nel documento con la relativa definizione (ed eventualmente un simbolo).

I pacchetti *acro* e *acronym* (se ne vedano le documentazioni), gestiscono efficacemente gli acronimi, generando automaticamente anche i collegamenti ipertestuali tra acronimo nel testo e relativa spiegazione nell'elenco.

Per comporre un vero e proprio glossario esiste il pacchetto *glossaries* (utile anche per gli acronimi), che tramite il programma *MakeIndex* genera automaticamente i corrispondenti elenchi.

5

MATEMATICA

Questo capitolo esplora uno dei principali punti di forza di \LaTeX , anche se ne intacca solamente la superficie: la composizione di formule matematiche. Si spiegheranno strumenti sufficienti per la *maggior parte* delle esigenze, ma potrebbe darsi che *quella particolare* necessità non trovi risposta in queste pagine. Se così fosse, molto probabilmente la soluzione sta in una delle funzioni del pacchetto `amsmath` (che non potrà essere descritto per intero, dati i limiti di questo lavoro) o di qualche altro pacchetto dedicato.

La scrittura della matematica e delle altre scienze è regolata da norme dipendenti dalle tradizioni e dalla cerchia dei lettori cui lo scritto è destinato, e raccolte nel mondo dalla normativa ISO (*International Standard Organization*, “Organizzazione Internazionale per lo Standard”), in Italia dalla normativa UNI (Ente Nazionale Italiano di Unificazione). Le norme emesse dall’UNI hanno valore di legge (per cui se un documento per uso legale viene scritto in modo conforme a esse non perde tale valore) ma nella versione originale le si può consultare soltanto tramite un servizio a pagamento.

Di qui in avanti si danno per caricati i pacchetti `amsmath` e `amssymb`.

5.1 FORMULE IN LINEA E IN DISPLAY

In linea generale, nelle formule matematiche le variabili vengono rese in corsivo matematico, diverso dal *corsivo ordinario* (si noti, però, che il font matematico \mathcal{AMS} Euler usato in questa guida è piuttosto simile al tondo, perché volutamente imita la scrittura diritta di un matematico alla lavagna).

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

5.1.1 Formule in linea

Una formula “in linea” è un’espressione che \LaTeX incorpora nel testo: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$, per esempio. Come si può osservare, il programma fa il *possibile* per comprimerla e modificare meno che può l’interlinea nel capoverso che la contiene. Non ci si preoccupi se con questa modalità di scrittura elementi che di solito si trovano sopra o sotto un simbolo gli compaiono accanto: è tipograficamente corretto. L’esempio seguente mostra come si scrive una formula di questo tipo:

Una formula in linea è un’espressione che `\LaTeX{}` incorpora nel testo:
`$$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$`. Il programma modifica il meno possibile l’interlinea del capoverso.

Una formula in linea è un’espressione che \LaTeX incorpora nel testo: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$. Il programma modifica il meno possibile l’interlinea del capoverso.

Le formule in linea si scrivono tra dollari $\$ \dots \$$ (oppure, ma meno spesso, tra i comandi $\backslash(\dots\backslash)$) e si consiglia di usarle solo con espressioni di altezza contenuta come le seguenti:

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

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

5.1.2 Formule in display

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

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

Come si può osservare, ora la formula è centrata, non compressa, e tutti i suoi elementi occupano il giusto spazio con un risultato finale di grande respiro.

L’unico modo corretto per scrivere queste formule è usare uno dei due ambienti matematici seguenti:

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

Degli altri modi esistenti per farlo, oggi *non devono essere più usati*:

- I dollari doppi `$$...$$`, che potrebbero compromettere la corretta spaziatura verticale delle formule o il funzionamento dell’opzione di classe `fleqn` [Fairbairns, 2012].
- Gli ambienti standard `eqnarray` e `eqnarray*` (per sistemi di formule numerate e non numerate rispettivamente), perché prima e dopo = inseriscono più spazio del dovuto. È un difetto conservato in $\text{\LaTeX}_{2\epsilon}$ per mantenerne la compatibilità con le vecchie versioni del programma.

Si considerino gli esempi seguenti:

Una formula in display è un’espressione che `\LaTeX` compone su linee a sé stanti:

```
\[
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2} =
\frac{\pi^2}{6}.
\]
```

Una formula in display è un’espressione che \LaTeX compone su linee a sé stanti:

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

Se f è continua e

```
\[
F(x) = \int_a^x f(t) dt,
\]
```

allora

```
\begin{equation}
F'(x) = f(x).
\end{equation}
```

Se f è continua e

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

allora

$$F'(x) = f(x). \quad (5.1)$$

Come si può osservare, le formule sono accompagnate da segni di punteggiatura.

Intorno alla punteggiatura nelle formule in display sono fiorite due scuole di pensiero. Alcuni ritengono che non andrebbe *mai* usata, perché superflua e causa di possibili ambiguità nella lettura [Beccari, 2012]. Altri, tra cui chi scrive, ritenendo le formule in genere parte integrante dell'argomentazione, pensano che la punteggiatura aiuti chi legge e che quindi ci vada [Guiggiani e Mori, 2008]. Qualunque delle due scuole si scelga, l'importante è seguirla in *tutto* il documento.

Si scrivono in display espressioni complesse e di grandi dimensioni (troppo sacrificate tra le righe di un capoverso) e formule più contenute a cui si voglia dare un risalto particolare.

I comandi `\label` ed `\eqref` permettono i riferimenti incrociati alle formule (come già visto nel paragrafo 3.10 a pagina 39):

<pre>\begin{equation} \label{eqn:eulero} e^{i\pi}+1=0. \end{equation} Dalla formula~\eqref{eqn:eulero} si deduce che\dots</pre>	$e^{i\pi} + 1 = 0. \quad (5.2)$ <p>Dalla formula (5.2) si deduce che...</p>
---	---

5.1.3 Modo matematico e modo testuale

La modalità con cui si scrive la matematica (*modo matematico*) differisce per alcuni aspetti da quella con cui si scrive il testo (*modo testuale*). Ecco i principali.

- \LaTeX inserisce automaticamente gli spazi in base alla struttura della formula e ignora quelli che trova nel sorgente (interruzioni di riga comprese). Se serve, si possono inserire *a mano* ulteriori spazi con i comandi raccolti nella tabella 25 a pagina 75.
- Nella scrittura delle formule *non sono ammesse righe vuote*.
- \LaTeX mette in corsivo matematico *tutte* le lettere che trova in una formula, considerandole altrettante variabili. Per inserire in una formula in display un (breve) testo in tondo e spaziato normalmente si usa il comando `\text`, esplicitando la spaziatura prima e dopo.

I due esempi seguenti mostrano quanto si è appena descritto:

<pre>\$x+y+z=n\$ \\ \$ x + y + z = n \$</pre>	$x + y + z = n$ $x + y + z = n$
<pre>\[z^2+1=0 \quad \quad \quad \text{per } z=\pm i. \]</pre>	$z^2 + 1 = 0 \quad \text{per } z = \pm i.$

Il comando `\pm` produce \pm (`\mp` produce \mp).

5.2 NOZIONI INTRODUTTIVE

Questo paragrafo descrive i comandi più usati per scrivere le formule matematiche. (Ulteriori comandi sono raccolti nelle tabelle contenute nelle prossime sezioni.)

5.2.1 Raggruppamenti

La maggior parte dei comandi matematici agisce soltanto sul carattere immediatamente successivo. Si evita questo comportamento racchiudendo il testo interessato in un gruppo di parentesi graffe:

<code>\[</code>	
<code>a^{x+y} \neq a^{x+y}</code>	$a^x + y \neq a^{x+y}$
<code>\]</code>	

5.2.2 Esponenti, indici e radici

Apici e pedici si scrivono dopo i caratteri `^` e `_` rispettivamente:

<code>\$a_1\$ \quad \$x^2\$ \quad \$e^{-\alpha t}\$ \quad \$a_{ij}^3\$</code>	$a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3$
<code>\$a^{3}_{ij}\$ \quad \$e^{x^2}\$ \quad \$e^{x^2}\$</code>	$e^{x^2} \neq e^{x^2}$

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

Gli indici di secondo ordine vanno messi in un gruppo di graffe insieme a quelli di ordine superiore: una scrittura come `x_{n_k}` non ha senso.

Dalla successione <code>\$x_n\$</code> estrarre <code>\$x_{n_k}\$</code> .	Dalla successione x_n estrarre x_{n_k} .
--	--

Il simbolo di radice quadrata si ottiene con `\sqrt`, quello di radice n-esima con

`\sqrt[n]{...}`

\LaTeX calcola automaticamente le dimensioni della radice:

<code>\$\sqrt{x}\$ \quad \$\sqrt{x^2+\sqrt{y}}\$ \quad \$\sqrt[3]{2}\$</code>	$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2}$
---	--

5.2.3 Somme, prodotti e frazioni

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

Trovare il massimo valore della funzione	Trovare il massimo valore della funzione
<code>\[</code>	
<code>f(x_1, \dots, x_n) = \prod_{k=1}^n x_k</code>	$f(x_1, \dots, x_n) = \prod_{k=1}^n x_k$
<code>\]</code>	
sotto la condizione	sotto la condizione
<code>\[</code>	
<code>\sum_{k=1}^n x_k^2 = 1.</code>	$\sum_{k=1}^n x_k^2 = 1.$
<code>\]</code>	

Una frazione, anche complessa, si ottiene semplicemente con il comando

`\frac{numeratore}{denominatore}`

Per piccole quantità di materiale frazionario, a volte la forma n/m è più gradevole sulla pagina:

<pre>\[\frac{x^{2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2} \]</pre>	$\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$
---	---

5.2.4 Limiti, derivate e integrali

Il comando

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

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

<pre>\[\lim_{x \to 0} \frac{\sin x}{x} = 1 \quad \lim_{n \to +\infty} f_n = \delta \]</pre>	$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \quad \lim_{n \rightarrow +\infty} f_n = \delta$
--	---

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

<pre>\[y=x^2, \quad y'=2x, \quad y''=2. \]</pre>	$y = x^2, \quad y' = 2x, \quad y'' = 2.$
---	--

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

<pre>\[\int_a^{a+T} f(x) \, dx = \int_0^T f(x) \, dx \]</pre>	$\int_a^{a+T} f(x) \, dx = \int_0^T f(x) \, dx$
--	---

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

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

<pre>\[\iint_D f(x,y) \, dx \, dy \quad \iiint g \, dx \, dy \, dz \]</pre>	$\iint_D f(x,y) \, dx \, dy \quad \iiint g \, dx \, dy \, dz$
--	---

Per gli integrali curvilinei lungo curve chiuse si usa `\oint`:

<pre>\[\oint f(z) \, dz = 2\pi i. \]</pre>	$\oint f(z) \, dz = 2\pi i.$
---	------------------------------

Tabella 20: Simboli insiemistici (`\bigcup` e `\bigcap` sono operatori).

\subset	<code>\subset</code>	\supset	<code>\supset</code>	\cup	<code>\cup</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cap	<code>\cap</code>
\in	<code>\in</code>	\ni o \owns	<code>\ni</code> o <code>\owns</code>	\notin	<code>\notin</code>
\complement	<code>\complement</code>	\setminus	<code>\setminus</code>	\emptyset	<code>\emptyset</code>
\bigcup	<code>\bigcup</code>	\bigcap	<code>\bigcap</code>		

5.2.5 Insiemi numerici

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

```
\[
x\in\mathbb{R}, \quad \quad \quad x \in \mathbb{R}, \quad z \in \mathbb{C}.
z\in\mathbb{C}.
\]
```

I simboli usati nell’esempio precedente sono raccolti insieme ad altri simboli insiemistici nella tabella 20.

Se si scrivono nel preambolo le definizioni seguenti (si veda il paragrafo 11.1 a pagina 173)

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

per avere \mathbb{N} basta scrivere `\N`, e si può cambiare notazione con un’unica modifica.

5.2.6 Lettere greche

Le lettere greche minuscole e maiuscole si ottengono con i comandi elencati nella tabella 21 nella pagina successiva (per l’*omicron* minuscolo si usa il carattere latino *o*, e per le maiuscole che non vi compaiono si usano le corrispondenti maiuscole latine, identiche a quelle greche). Per le sei di esse prefissate con `var-` si noti quanto segue:

- `\varpi` e `\varsigma` non si usano *mai*;
- le altre quattro vanno usate in modo esclusivo: *o* la forma principale *o* la sua variante (ma si ricordi che a differenza di altri Paesi, in Europa si usa generalmente la seconda).

È per questo motivo che anche se nel font matematico in uso le diverse forme fossero uguali (come accade in questa guida) non ci sarebbe pericolo di confondersi o di commettere errori.

Diventa perciò conveniente ridefinire queste quattro varianti come caratteri normali, scrivendo nel preambolo (si veda il paragrafo 11.1 a pagina 173):

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

Tabella 21: Lettere greche (il font matematico \mathcal{AMS} Euler usato in questa guida non permette di distinguere tra le diverse forme minuscole di ρ e σ).

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

5.2.7 Simboli che sormontano altri simboli

Il comando

```
\overset{⟨primo argomento⟩}{⟨secondo argomento⟩}
```

produce il simbolo indicato nel $\langle \text{primo argomento} \rangle$ rimpicciolito e sovrapposto a quello scritto nel $\langle \text{secondo argomento} \rangle$ (di solito un simbolo di relazione binaria), che rimane delle sue dimensioni e nella posizione abituale. Il comando `\underset` fa l'opposto.

Il simbolo

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

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

Il simbolo

$$\underline{H}$$

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

5.2.8 Barre e accenti

Il comando `\bar` pone un trattino sul proprio argomento: il simbolo \bar{x} indica un nome di variabile distinto da x (si veda la tabella 22 nella pagina successiva).

I comandi `\overline` e `\underline` (il secondo dei quali si usa piuttosto raramente) sovrapposono e sottolineano rispettivamente tutto il proprio argomento: il simbolo \overline{x} indica un operatore applicato alla variabile x :

```

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

```

I comandi `\vec` e `\overrightarrow` agiscono esattamente come `\bar` e `\overline`, ma producono frecce anziché barre orizzontali.

```

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

```

Esistono tre tipi di barra verticale, distinguibili dallo spazio richiesto prima e dopo:

- semplice | (ottenibile anche con `\vert`);

Tabella 22: Accenti in modo matematico.

\bar{a}	<code>\bar{a}</code>	\hat{a}	<code>\hat{a}</code>	\tilde{a}	<code>\tilde{a}</code>
\vec{a}	<code>\vec{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>
\check{a}	<code>\check{a}</code>	\widehat{abc}	<code>\widehat{abc}</code>	\widetilde{abc}	<code>\widetilde{abc}</code>

- delimitatore sinistro e destro (`\lvert` e `\rvert` rispettivamente);
- relazione binaria (`\mid`) per la divisibilità e il *tale che* negli insiemi.

<code>\$F(x) _{x=\gamma(t)}\$ \quadquad</code>	$F(x) _{x=\gamma(t)}$	$ x $
<code>\$\lvert x\rvert\$ \quad[1ex]</code>		
Se <code>\$p\mid n^2\$</code> ,	Se $p \mid n^2$,	allora $p \mid n$.
allora <code>\$p\mid n\$</code> .		

I comandi appena esaminati prevedono forme analoghe per barre verticali doppie: `\|` (o `\Vert`), `\lVert`, `\rVert` e `\parallel`.

La barra laterale è prevista talvolta nel calcolo degli integrali:

<code>\[</code>		
<code>\int_a^b f(x)\,dx = F(x)\big _a^b</code>	$\int_a^b f(x) dx = F(x) _a^b$	
<code>\]</code>		

La sua altezza va regolata a mano, premettendole uno dei comandi che verranno descritti nel paragrafo 5.4 a pagina 77.

La differenza tra due insiemi si realizza con il comando `\setminus`. Si confrontino le due scritture seguenti:

<code>\$A\backslash B\$ \quad \backslash</code>	$A \setminus B$
<code>\$A\setminus B\$</code>	$A \setminus B$

Come si può osservare, il primo codice produce una spaziatura (leggermente) sbagliata.

Per il valore assoluto e la norma conviene caricare il pacchetto `mathtools` e definire nel preambolo due comandi ad hoc:

```
\DeclarePairedDelimiter{\abs}{\lvert}{\rvert}
\DeclarePairedDelimiter{\norma}{\lVert}{\rVert}
```

da usare nel modo seguente:

<code>\[</code>		
<code>\sum_{n=0}^{+\infty} z^n =</code>	$\sum_{n=0}^{+\infty} z^n =$	$\frac{1}{1-z}$ per $ z < 1$.
<code>\frac{1}{1-z} \quadquad</code>		
<code>\text{per } \abs{z} < 1.</code>		
<code>\]</code>		

<code>\[</code>		
<code>\norma{x} =</code>	$\ x\ =$	$\sqrt{x_1^2 + \dots + x_n^2}$
<code>\sqrt{x_1^2 + \dots + x_n^2}</code>		
<code>\]</code>		

Le varianti asterisco dei comandi appena definiti producono delimitatori ad altezza variabile:

<code>\[</code>		
<code>\abs*{\frac{a}{b}} \quadquad</code>	$\left \frac{a}{b} \right $	$\left\ \frac{u}{\lambda} \right\ $
<code>\norma*{\frac{u}{\lambda}}</code>		
<code>\]</code>		

Tabella 23: Freccie.

\leftarrow	<code>\leftarrow</code> o <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>
\rightarrow	<code>\rightarrow</code> o <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>	\Leftrightarrow	<code>\Leftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>
\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>

Esiste un'ulteriore variante per entrambi i comandi:

```
\abs[⟨grandezza del delimitatore⟩]{⟨testo⟩}
```

e

```
\norma[⟨grandezza del delimitatore⟩]{⟨testo⟩}
```

che permette di regolare a mano la grandezza dei delimitatori specificandola nell'argomento facoltativo. Si consideri l'esempio seguente:

<pre>\[\abs[\bigg]{\sum_{i=1}^n x_i} \qquad \abs*{\sum_{i=1}^n x_i} \]</pre>	$\left \sum_{i=1}^n x_i \right \quad \left \sum_{i=1}^n x_i \right $
---	---

Come si può osservare, la prima scrittura presenta delimitatori (leggermente) insufficienti.

Per aggiungere alle variabili un accento matematico, come un cappello o una tilde, si possono usare i comandi della tabella 22 nella pagina precedente. I comandi `\widehat` e `\widetilde` producono rispettivamente simboli di cappello e tilde che sormontano tutto il proprio argomento.

5.2.9 Punti, frecce e simboli logici

In matematica esistono due tipi di due punti, distinguibili dal diverso spazio richiesto prima e dopo:

- semplice `:`, spaziato come in un'operazione binaria (divisione);
- `\colon`, spaziato come un segno di interpunzione.

Si confrontino le due scritture seguenti:

<code>\$f:\mathbb{R}\to\mathbb{R}\$</code>	$f : \mathbb{R} \rightarrow \mathbb{R}$
<code>\$f\colon\mathbb{R}\to\mathbb{R}\$</code>	$f : \mathbb{R} \rightarrow \mathbb{R}$

Come si può osservare, il primo codice produce una spaziatura (leggermente) sbagliata.

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

Tabella 24: Simboli logici.

\vee	<code>\lor</code>	\wedge	<code>\land</code>	\neg	<code>\neg</code>
\exists	<code>\exists</code>	\nexists	<code>\nexists</code>	\forall	<code>\forall</code>
\implies	<code>\implies</code>	\iff	<code>\iff</code>	\models	<code>\models</code>

```

\l
x_1,\dots,x_n \quad \quad \quad x_1,\dots,x_n \quad x_1 + \dots + x_n
x_1+\dots+x_n
\r

```

Per ulteriori esempi dell'uso di `\dots`, si rimanda al paragrafo 5.5 a pagina 79.

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

```

$f\colon\mathbb{R}\to\mathbb{R}$ \quad \quad \quad f: \mathbb{R} \rightarrow \mathbb{R}
$x\mapsto x^2$ \quad \quad \quad x \mapsto x^2

```

I comandi `\xleftarrow` e `\xrightarrow` producono frecce che si estendono automaticamente per accordarsi con indici di lunghezza non comune. Entrambi accettano un argomento facoltativo (il pedice) e uno obbligatorio (l'apice) che possono anche rimanere vuoti.

```

\l
\xleftarrow{a} \quad \quad \quad \xleftarrow{a}
\xleftarrow[X]{a+b} \quad \quad \quad \xleftarrow[X]{a+b}
\xrightarrow[X+Y+Z]{} \quad \quad \quad \xrightarrow[X+Y+Z]{}
\r

```

La tabella 23 nella pagina precedente raccoglie i principali comandi per ottenere le frecce.

I simboli logici raccolti nella tabella 24 vanno usati *solo* in un lavoro sull'argomento (mentre in tutti gli altri contesti vanno senz'altro preferite le forme estese *se... allora, se e solo se, eccetera*).

5.2.10 Spazi in modo matematico

Può accadere, anche se di rado, che la spaziatura scelta da \LaTeX per le formule risulti insoddisfacente. Per modificarla si usano i comandi raccolti nella tabella 25 a fronte. L'unità di misura dello spazio matematico è il *quadrato* (`\quad`), il cui valore standard (1 em) varia in base al punto dell'espressione matematica in cui viene dato.

Spazi troppo stretti

Lo spazio sottile prodotto dal comando `\,`, è molto utile in alcune formule. Si confronti

```

\l
\int_a^b f(x) dx, \quad \quad \quad \int_a^b f(x) dx, \quad \sqrt{2}a, \quad \sqrt{\log x}.
\sqrt{2} a, \quad \quad \quad
\sqrt{\log x}.
\r

```

Tabella 25: Spazi in modo matematico.

Comando	Tipo di spazio
<code>\,</code>	Spazio sottile positivo
<code>\!</code>	Spazio sottile negativo
<code>\quad</code>	Spazio di un <i>quadrato</i>
<code>\qquad</code>	Spazio di un <i>quadrato</i> ne

con

<pre>\[\int_a^b f(x)\,dx, \quad \sqrt{2}\,a, \quad \sqrt{\log x}. \]</pre>	$\int_a^b f(x) dx, \quad \sqrt{2}a, \quad \sqrt{\log x}.$
---	---

Come si può osservare, il primo codice produce una spaziatura (leggermente) insufficiente.

Spazi troppo larghi

Il comando `\!` può migliorare le spaziature eccessive prodotte, per esempio, da simboli inclinati (barre di frazione, radicali) preceduti o seguiti da simboli particolari (esponenti, apici o operatori). Si confronti

<pre>\[x^2/2, \quad a/\sin b. \]</pre>	$x^2/2, \quad a/\sin b.$
---	--------------------------

con

<pre>\[x^2\!/2, \quad a/\!\sin b. \]</pre>	$x^2/2, \quad a/\sin b.$
---	--------------------------

Come si può osservare, il primo codice produce una spaziatura (leggermente) esagerata.

Si ricorda che i due spazi appena descritti vanno usati per migliorare ulteriormente la qualità (già alta) del documento *se e solo se* la spaziatura predefinita non fosse adeguata ai simboli che compaiono nelle formule.

5.3 OPERATORI

5.3.1 Caratteristiche generali

In \LaTeX , le funzioni come `sin`, `cos` e `log` presentano le seguenti caratteristiche:

- per essere più visibili sulla pagina (in accordo con le norme ISO-UNI) vengono rese in tondo normale e non in corsivo matematico come le variabili;
- richiedono una particolare spaziatura prima e dopo, che il programma inserisce automaticamente;

Tabella 26: Operatori predefiniti.

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

Se il nuovo operatore presenta indici scritti come nei limiti (lim, sup o max), si usa la variante asterisco di `\DeclareMathOperator`:

```
\DeclareMathOperator*{\argmax}{arg\,max}
```

5.4 PARENTESI

\LaTeX e `amsmath` definiscono numerosi simboli per parentesi e altri delimitatori. Le parentesi tonde e quadre si scrivono con i corrispondenti caratteri da tastiera, mentre quelle graffe *anche in modo matematico* devono essere precedute da `\`. Tutti gli altri delimitatori vengono generati da comandi dedicati.

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

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

Talvolta bisogna aggiustarne a mano le dimensioni: lo si può fare prefissandoli con i comandi `\big`, `\Big`, `\bigg` e `\Bigg`. I comandi `\bigl` (*big left*) e `\bigr` (*big right*) ingrandiscono lievemente le parentesi:

```
\[
\bigl( (x-y)+(x+y) \bigr)
\]
```

$$((x - y) + (x + y))$$

I comandi `\Bigl` e `\Bigr` producono parentesi ancora più grandi:

```
\[
\Bigl(1+\frac{1}{n}\Bigr)^n
\]
```

$$\left(1 + \frac{1}{n}\right)^n$$

I comandi `\biggl` e `\biggr` ne generano di più grandi ancora:

```
\[
\biggl(\sum_n x_n^2\biggr)^{1/2}
\]
```

$$\left(\sum_n x_n^2\right)^{1/2}$$

Se non basta, ci sono anche `\Biggl` e `\Biggr`.

```
\$ \bigl( \Bigl( \biggl( \Biggl( \$
\quad
\biggr) \Biggr) \biggr) \Bigl) \$
\quad
\bigl| \Bigl| \biggl| \Biggl| \$
```

$$\left(\left(\left(\left(\right)\right)\right)\right) \quad |||||$$

Si noti che alcuni visualizzatori di PDF potrebbero mostrare non correttamente *a schermo* delimitatori particolarmente grandi. Il problema non si presenta, invece, a stampa.

5.5 VETTORI E MATRICI

I vettori si scrivono di solito in tondo nero (corsivo, secondo le norme ISO-UNI) oppure in semplice corsivo matematico; talvolta, soprattutto nei testi di fisica, sono sormontati da una freccia. Nel primo caso si può usare il comando `\mathbf`; nel secondo il comando `\bm` del pacchetto `bm`; nel terzo il comando `\vec`. Può essere conveniente ridefinire nel preambolo quest'ultimo comando (si veda il paragrafo 11.1 a pagina 173):

```
\renewcommand{\vec}{\bm}
```

In questo modo basta scrivere `\vec{v}` per ottenere \mathbf{v} e si può cambiare notazione con un'unica modifica (si veda anche il paragrafo 5.8 a pagina 83).

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

Gli elementi della matrice vengono centrati automaticamente, e righe e colonne si scrivono come una normale tabella `tabular`, ricordando che gli spazi espliciti sono ignorati.

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

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

```
\[
\begin{vmatrix}
1-x & 2 \\
3 & 4-x
\end{vmatrix}
\end{vmatrix}
\]
```

Si possono scrivere matrici con punti ellittici, come nell'esempio seguente:

```
\[
A=
\begin{bmatrix}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{bmatrix}
\end{bmatrix}
\]
```

Come si può osservare, il comando `\vdots` produce tre punti ellittici verticali e `\ddots` tre in diagonale.

Il comando

```
\hdotsfor{n}
```

Tabella 27: Simboli misti (`\bigodot`, `\bigoplus` e `\bigotimes` sono operatori; il font matematico \mathcal{AMS} Euler usato in questa guida rende allo stesso modo `\hbar` e `\hslash`).

\vee	<code>\vee</code>	\wedge	<code>\wedge</code>	\div	<code>\div</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>
∂	<code>\partial</code>	∇	<code>\nabla</code>	\cdot	<code>\cdot</code>
\hbar	<code>\hbar</code>	\hslash	<code>\hslash</code>	\circ	<code>\circ</code>
\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	\bullet	<code>\bullet</code>
ℓ	<code>\ell</code>	\wp	<code>\wp</code>	$\sqrt{\quad}$	<code>\sqrt</code>
\dagger	<code>\dagger</code>	\ddagger	<code>\ddagger</code>	$*$	<code>\ast</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\angle	<code>\angle</code>
\triangleleft	<code>\triangleleft</code>	\triangleright	<code>\triangleright</code>	\triangle	<code>\triangle</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\diamond	<code>\diamond</code>
\odot	<code>\odot</code>	\oplus	<code>\oplus</code>	\otimes	<code>\otimes</code>
\bigodot	<code>\bigodot</code>	\bigoplus	<code>\bigoplus</code>	\bigotimes	<code>\bigotimes</code>

riempie di punti la riga della matrice per $\langle n \rangle$ colonne:

```
\[
\begin{bmatrix}
a_{11} & \dots & a_{1n} \\
a_{21} & \dots & a_{2n} \\
\hdotsfor{3} \\
a_{n1} & \dots & a_{nn}
\end{bmatrix}
\end{bmatrix}
```

Una piccola matrice in linea si scrive nell'ambiente `smallmatrix`:

```
Sia $A=\bigl(
\begin{smallmatrix}
a & b \\
c & d
\end{smallmatrix}
\biggr)$
una matrice invertibile.
```

Sia $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ una matrice invertibile.

In questo caso le parentesi vanno aggiunte a mano.

La tabella 27 raccoglie altri simboli matematici di uso comune. Per un loro elenco completo si veda [Pakin, 2009].

5.6 SPEZZARE FORMULE LUNGHE

\LaTeX non spezza automaticamente una formula più lunga d'una riga. Solo chi l'ha scritta, infatti, ne conosce il ritmo di lettura e sa dov'è più opportuno andare a capo e se allinearne o meno le varie parti. In generale, tuttavia, vale la regola per cui si può andare a capo:

- dopo i simboli di relazione e, subordinatamente, dopo i simboli di relazione e di operazione binaria (si veda la tabella 28) nelle formule in linea;
- prima dei simboli di relazione e dopo i simboli binari nelle formule in display;

Tabella 28: Simboli di relazione.

\leq	<code>\le</code>	\geq	<code>\ge</code>	\sim	<code>\sim</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\simeq	<code>\simeq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\approx	<code>\approx</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\asymp	<code>\asymp</code>
\parallel	<code>\parallel</code>	\perp	<code>\perp</code>	\cong	<code>\cong</code>
\mid	<code>\mid</code>	\propto	<code>\propto</code>	\equiv	<code>\equiv</code>
\neq	<code>\neq</code>				

- *mai* dopo gli operatori funzionali (si veda la tabella 26 a pagina 77), i grandi operatori e i delimitatori di apertura.

Per spezzare e raggruppare formule in `display`, il pacchetto `amsmath` definisce (fra gli altri) gli ambienti `multline`, `split`, `gather` e `align`, che si descrivono di seguito.

5.6.1 Spezzare formule senza incolonnarle: `multline`

Per spezzare una formula in più righe non incolonnate si usa l'ambiente `multline`.

<pre>\begin{multline} f=a+b+c \\ +d+e+g+h \\ +r+s+t. \end{multline}</pre>	$ \begin{aligned} f &= a + b + c \\ &+ d + e + g + h \\ &+ r + s + t. \quad (5.3) \end{aligned} $
---	---

Si noti che:

- la prima riga viene allineata a sinistra e l'ultima a destra;
- le rimanenti vengono centrate (a meno che non sia attiva l'opzione `fleqn`, che allinea comunque le formule a sinistra rispetto a un margine rientrato);
- il numero progressivo della formula viene messo nel margine destro in corrispondenza dell'ultima riga.

La variante asterisco `multline*` produce formule dello stesso tipo *non numerate*.

5.6.2 Spezzare formule incolonnandole: `split`

Per spezzare una formula in più righe incolonnate si usa l'ambiente `split`.

<pre>\begin{equation} \begin{split} a &= b+c-d \\ &= e-f \\ &= g+h \\ &= i. \end{split} \end{equation}</pre>	$ \begin{aligned} a &= b + c - d \\ &= e - f \\ &= g + h \\ &= i. \quad (5.4) \end{aligned} $
--	---

Come si può osservare:

- il carattere `&` incolonna le righe della formula a partire dal punto in cui viene dato (di solito subito prima di un `=`);

- `split` va *necessariamente* usato dentro un altro ambiente per la matematica in `display`, responsabile della numerazione della formula (ma si può ottenere una formula di questo tipo anche non numerata).

5.7 RAGGRUPPARE PIÙ FORMULE

Per raggruppare più formule in `display`, il pacchetto `amsmath` definisce (fra gli altri) gli ambienti `gather` e `align`, descritti di seguito.

5.7.1 Raggruppare formule senza incolonnarle: `gather`

L'ambiente `gather` raggruppa più formule centrandole e numerando autonomamente ciascuna su una riga a sé e, se necessario, assegnandole un'etichetta.

```
\begin{gather}
a=b+c, \\
V+F-S=2.
\end{gather}
```

$$a = b + c, \quad (5.5)$$

$$V + F - S = 2. \quad (5.6)$$

Si noti che si avrebbe un risultato “simile” scrivendo ogni formula in un ambiente `equation`, ma poi lo spazio tra di esse sarebbe esagerato. La variante asterisco `gather*` produce formule dello stesso tipo non numerate.

5.7.2 Raggruppare formule incolonnandole: `align`

L'ambiente `align` incolonna gruppi di due o più formule mettendo e numerando ciascuna su una riga a sé, come mostra l'esempio seguente:

```
\begin{align}
a &= b+c+d, \\
e &= f, \\
x-1 &= y+z.
\end{align}
```

$$a = b + c + d, \quad (5.7)$$

$$e = f, \quad (5.8)$$

$$x - 1 = y + z.$$

La variante asterisco `align*` produce formule dello stesso tipo non numerate. (Si ottiene lo stesso risultato dando `\notag` alla fine della formula interessata).

L'ambiente è utile anche per incolonnare più righe di formule autonome. In tal caso, `&` assume due significati diversi a seconda della posizione in cui si trova sulla riga:

- se occupa un posto dispari, indica a \LaTeX i punti da incolonnare;
- se occupa un posto pari, è un separatore come in `tabular`.

```
\begin{align}
a &= b, & c &= d, & e &= f, \\
u &= v, & w &= x, & y &= z.
\end{align}
```

$$a = b, \quad c = d, \quad e = f, \quad (5.9)$$

$$u = v, \quad w = x, \quad y = z. \quad (5.10)$$

5.7.3 Gli ambienti `gathered` e `aligned`

Per raggruppare più formule con una parentesi, magari da integrare con del testo esplicativo, \LaTeX definisce i due ambienti `gathered` e `aligned`, che trattano le espressioni al proprio interno come i due ambienti corrispondenti appena esaminati.

```

\left.
\begin{aligned}
a &= b+1 \\
c &= d
\end{aligned}
\right\}
\quad
\text{due equazioni}

```

$$\left. \begin{array}{l} a = b + 1 \\ c = d \end{array} \right\} \text{ due equazioni}$$

5.7.4 Casi e sottonumerazioni

L'ambiente `cases` serve per le definizioni fatte per casi. Graffa e allineamento sono automatici; il testo nella seconda colonna va nell'argomento di `\text`.

```

\begin{cases}
1, & \text{se } n=0, \\
n(n-1)!, & \text{se } n \geq 1.
\end{cases}

```

$$n! = \begin{cases} 1, & \text{se } n = 0, \\ n(n-1)!, & \text{se } n \geq 1. \end{cases}$$

L'ambiente `subequations` produce sottonumerazioni:

```

\begin{subequations}
\label{eqn:schema}
\begin{aligned}
a &= b+c, \\
c &= d, \\
e &= f+g.
\end{aligned}
\end{subequations}
Le formule~\eqref{eqn:schema},
e in particolare
la~\eqref{eqn:sub}, \dots

```

$$a = b + c, \quad (5.11a)$$

$$c = d, \quad (5.11b)$$

$$e = f + g. \quad (5.11c)$$

Le formule (5.11), e in particolare la (5.11b), ...

5.8 MODIFICARE STILE E CORPO DEL FONT

In modo matematico, \LaTeX armonizza stile e corpo del font con il contesto in cui le formule si trovano. A volte, però, può essere necessario modificare a mano questi due parametri: nei prossimi paragrafi si spiega come farlo.

Modificare lo stile

Gli esempi seguenti mostrano come i comandi per cambiare lo stile del font (raccolti anche nella tabella 29 nella pagina successiva) agiscono su lettere e numeri, ma *non* sui segni di operazione.

```

$x+y+2^n M \cos t$ \\
$\mathit{x+y+2^n M \cos t}$ \\
$\mathbf{x+y+2^n M \cos t}$ \\
$\mathrm{x+y+2^n M \cos t}$ \\
$\mathhtt{x+y+2^n M \cos t}$ \\
$\mathsf{x+y+2^n M \cos t}$

```

```

x + y + 2^n M cos t

```

Tabella 29: Stili dei font matematici (`\mathscr` richiede il pacchetto `mathrsfs`, `\mathbb` e `\mathfrak` il pacchetto `amsfonts` o `amssymb`). Si noti che alcuni di essi non hanno effetto su minuscole e numeri.

Stile	Codice	Risultato
Tondo	<code>\mathrm{ABCdef123}</code>	ABCdef123
Corsivo	<code>\mathit{ABCdef123}</code>	<i>ABCdef123</i>
Nero	<code>\mathbf{ABCdef123}</code>	ABCdef123
Dattilografico	<code>\mathtt{ABCdef123}</code>	ABCdef123
Senza grazie	<code>\mathsf{ABCdef123}</code>	ABCdef123
Gotico	<code>\mathfrak{ABCdef123}</code>	℔℔℔def123
Nero da lavagna	<code>\mathbb{ABC}</code>	℔ABC
Calligrafico	<code>\mathcal{ABC}</code>	<i>ABC</i>
Manoscritto	<code>\mathscr{ABC}</code>	<i>ABC</i>

Spesso i principianti abusano dei simboli in nero, tipograficamente piuttosto “pesanti”. Questi si ottengono con il comando `\mathbf`, il cui comportamento dipende in larga parte dalla serie di font matematici in uso e di solito ha effetto solo sulle lettere. Si osservi l’esempio seguente:

```

\l
\mu, M \quad
\mathbf{\mu}, \mathbf{M}
\l

```

$\mu, M \quad \mathbf{\mu}, \mathbf{M}$

Come si può osservare, il comando non ha effetto sulla lettera μ e produce lettere in tondo anziché in corsivo matematico.

Per comporre simboli matematici in nero corsivo si consiglia il comando `\bm` del pacchetto `bm`. Si tenga presente che `\bm` funziona *solo se* il font matematico corrente dispone della versione nera di quel simbolo.

```

\l
\mu, M \quad
\bm{\mu}, \bm{M}
\l

```

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

Gli indici letterali vanno scritti in corsivo matematico se rappresentano quantità variabili (cioè se sono *simboli*), in tondo se rappresentano apposizioni di una grandezza fisica (cioè se sono semplice *testo*). In quest’ultimo caso si usa il comando `\textup` (o, in alternativa, il comando `\ped` definito dall’opzione `italian` di `babel`, si veda il paragrafo 4.3.2 a pagina 53):

```

\l
V_\textup{eff} \quad
\psi\ped{incidente}
\l

```

$V_{\text{eff}} \quad \psi_{\text{incidente}}$

Si confrontino le due scritture seguenti:

```

$V_{\text{eff}}$ \quad V_{\text{eff}}
$V_\textup{eff}$ \quad V_{\text{eff}}

```

Come si può osservare, \LaTeX interpreta il pedice nel primo codice come tre variabili da moltiplicare, rendendole in corsivo matematico e spaziandole di conseguenza. Il codice corretto è il secondo.

Modificare il corpo

In modo matematico si può impostare a mano la dimensione del font con le quattro dichiarazioni `\displaystyle`, `\textstyle`, `\scriptstyle` e `\scriptscriptstyle`. Si considerino gli esempi seguenti:

```
\[
\sum_{k=1}^n z^k \quad
\textstyle\sum_{k=1}^n z^k
\]
\displaystyle\sum_{k=1}^n z^k$
\scriptstyle\sum_{k=1}^n z^k$
\scriptscriptstyle
\sum_{k=1}^n z^k$
```

$$\sum_{k=1}^n z^k \quad \sum_{k=1}^n z^k$$

$$\sum_{k=1}^n z^k \quad \sum_{k=1}^n z^k \quad \sum_{k=1}^n z^k \quad \sum_{k=1}^n z^k$$

```
\[
x_G=
\frac{\displaystyle
\sum_{i=1}^n m_i x_i}
{\displaystyle\sum_{i=1}^n m_i}
\]
```

$$x_G = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}$$

Come si può osservare, il cambiamento del corpo influisce anche sul modo in cui vengono resi gli indici.

Si possono usare i due comandi `\dfrac` e `\tfrac` come abbreviazioni di `\displaystyle\frac{...}` e `\textstyle\frac{...}`:

```
\frac{1}{k}\log_2 c(f)$
\quad
\dfrac{1}{k}\log_2 c(f)$
\[
\frac{1}{k}\log_2 c(f) \quad
\tfrac{1}{k}\log_2 c(f)
\]
```

$$\frac{1}{k} \log_2 c(f) \quad \frac{1}{k} \log_2 c(f)$$

$$\frac{1}{k} \log_2 c(f) \quad \frac{1}{k} \log_2 c(f)$$

5.9 EVIDENZIARE FORMULE

Per evidenziare un'intera formula con uno sfondo colorato o con una cornice è utile il pacchetto `empheq`. (Il primo dei due esempi che seguono richiede il pacchetto `xcolor`.)

```
\newcommand*\mygraybox[1]{%
\colorbox{lightgray}{#1}}
\begin{empheq}[box=%
\mygraybox]{align*}
a &= b \\
E &= mc^2 + \int_a^a x \, dx
\end{empheq}
```

$$a = b$$

$$E = mc^2 + \int_a^a x \, dx$$

```
\begin{empheq}[box=%
\fbox]{align*}
a &= b \\
E &= mc^2 + \int_a^a x \, dx
\end{empheq}
```

$$a = b$$

$$E = mc^2 + \int_a^a x \, dx$$

Per evidenziare solo una parte di una formula (anche in una presentazione), c'è il pacchetto `hf-tikz`.

5.10 ENUNCIATI E DIMOSTRAZIONI

Di qui in avanti si dà per caricato *anche* il pacchetto `amsthm`.

5.10.1 Enunciati

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

- stabilire il tipo di enunciato da inserire (per esempio, definizioni e teoremi);
- assegnare un nome a ogni ambiente (per esempio, *definizione* e *teorema*; non si può usare `def`, perché è già un comando di base di \LaTeX);
- titolare gli enunciati (per esempio, con *Definizione* e *Teorema*).

Il comando `\newtheorem`, dato *nel preambolo*, permette di fare le relative dichiarazioni globali e prevede due forme di definizione:

```
\newtheorem{<nome dell'enunciato>}{<titolo>}[<sezione>]
```

oppure, in alternativa,

```
\newtheorem{<nome dell'enunciato>}[<numerato come>]{<titolo>}
```

dove:

- `<nome dell'enunciato>` è una parola chiave che identifica l'enunciato;
- `<titolo>` specifica il titolo dell'enunciato che comparirà nel documento;
- `<sezione>` specifica il livello di sezionamento (`chapter` o `section`, di regola) a cui collegare la numerazione dell'enunciato;
- in `<numerato come>` si scrive il nome di un enunciato dichiarato in precedenza, in modo che quello nuovo ne prosegua la numerazione.

La variante asterisco `\newtheorem*` produce enunciati non numerati.

Il testo dell'enunciato va messo nel corrispondente ambiente, e un'eventuale specificazione (fra parentesi tonde, nel documento finito) si scrive nell'argomento facoltativo immediatamente *dopo* il comando d'apertura, così:

```
\begin{<nome dell'enunciato>}[<eventuale specificazione>]
...
\end{<nome dell'enunciato>}
```

Il pacchetto `amsthm` prevede tre stili predefiniti per gli enunciati (`plain`, `definition` e `remark`) i cui dettagli tipografici dipendono dalla classe di documento in uso, anche se in linea di massima il primo dei tre produce il proprio contenuto in corsivo, mentre gli altri due lo lasciano in tondo. Di seguito si riportano le principali categorie di enunciato, ciascuna associata al proprio stile più tipico:

plain Per teoremi, lemmi, corollari, proposizioni, congetture, criteri, leggi e algoritmi.

definition Per definizioni, condizioni, problemi ed esempi.

remark Per osservazioni e annotazioni.

Si possono scrivere gli enunciati con stili diversi dal predefinito `plain` dividendoli in gruppi per tipo di enunciato e premettendo a ciascun gruppo il comando `\theoremstyle`. (Si può personalizzare lo stile dell'enunciato con il comando `\newtheoremstyle`.)

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

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

e

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

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

```
\begin{definizione}[di Gauss]
Si dice \emph{matematico} colui
per il quale è ovvio che
 $\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}$ .
\end{definizione}
\begin{teorema}
I matematici, se ce ne sono,
sono molto rari.
\end{teorema}
```

Definizione 1 (di Gauss). Si dice *matematico* colui per il quale è ovvio che $\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}$.

Teorema 1. *I matematici, se ce ne sono, sono molto rari.*

```
Il seguente teorema è a tutti
ben noto.
\begin{teorema}[di Pitagora]
In un triangolo rettangolo,
la somma dei quadrati costruiti
sui cateti è uguale
al quadrato costruito
sull'ipotenusa.
\end{teorema}
La dimostrazione è lasciata
per esercizio.
```

Il seguente teorema è a tutti ben noto.

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

La dimostrazione è lasciata per esercizio.

Come si può osservare, \LaTeX :

- produce etichetta e numero (automatico) dell'enunciato in tondo nero e la conclude con un punto fermo;
- separa ogni enunciato dal resto del testo senza rientrarlo;
- mette le definizioni in tondo e i teoremi in corsivo.

L'esempio seguente riguarda la numerazione di tre enunciati consecutivi, il primo e il terzo dei quali sono dello stesso tipo, ma il secondo no. Scrivendo nel preambolo

```
\theoremstyle{plain}
\newtheorem{legge}{Legge}
\newtheorem{decreto}[legge]{Decreto}
```

gli ambienti `legge` e `decreto` si usano come segue:

<pre>\begin{legge} \label{lex:capo} Il capo ha ragione. \end{legge} \begin{decreto}[Aggiornamento alla legge~\ref{lex:capo}] Il capo ha \emph{sempre} ragione. \end{decreto} \begin{legge} Se il capo ha torto, vedere la legge~\ref{lex:capo}. \end{legge}</pre>	<p>Legge 1. <i>Il capo ha ragione.</i></p> <p>Decreto 2 (Aggiornamento alla legge 1). <i>Il capo ha sempre ragione.</i></p> <p>Legge 3. <i>Se il capo ha torto, vedere la legge 1.</i></p>
---	---

Si osservi che il numero assegnato a *Decreto* prosegue la numerazione di *Legge* anziché cominciarne una nuova, perché l'argomento facoltativo [`legge`] nella definizione del teorema assegna entrambi gli enunciati allo stesso contatore. Come di consueto, `\label` permette riferimenti incrociati anche a enunciati matematici.

Se si desidera introdurre un enunciato *Murphy*, per esempio, la cui numerazione sia collegata al paragrafo corrente, basta specificare nell'argomento facoltativo l'opzione `section`, in questo modo:

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

L'ambiente `murphy` così definito si usa come al solito:

<pre>\begin{murphy} Se esistono due o più modi per fare una cosa, e uno di questi può creare una catastrofe, allora qualcuno lo sceglierà. \end{murphy}</pre>	<p>Murphy 5.10.1. <i>Se esistono due o più modi per fare una cosa, e uno di questi può creare una catastrofe, allora qualcuno lo sceglierà.</i></p>
---	--

5.10.2 Dimostrazioni

L'ambiente `proof` permette di scrivere una dimostrazione, che nel documento finito sarà chiusa da un quadratino. Si osservi l'esempio seguente:

<pre>\begin{teorema}[Sorpresa] Si ha che \$\log(-1)=0\$. \end{teorema} \begin{proof} Si ha che \$\log(-1)^2=2\log(-1)\$. Ma è anche vero che \$\log(-1)^2=\log(1)=0\$. Perciò \$\log(-1)=0\$, da cui la tesi. \end{proof}</pre>	<p>Teorema 3 (Sorpresa). <i>Si ha che $\log(-1) = 0$.</i></p> <p><i>Dimostrazione.</i> Si ha che $\log(-1)^2 = 2\log(-1)$. Ma è anche vero che $\log(-1)^2 = \log(1) = 0$. Perciò $2\log(-1) = 0$, da cui la tesi. \square</p>
---	--

Per sostituire la scritta *Dimostrazione* con un'altra, per esempio *Soluzione*, basta scrivere

```
\begin{proof}[Soluzione]
```

Il comando `\qedhere` colloca correttamente il simbolo di fine dimostrazione anche se questa termina con una formula in display. Si confronti

```
\begin{proof}
Basta usare la formula
\[
E=mc^2.
\]
\end{proof}
```

Dimostrazione. Basta usare la formula

$$E = mc^2.$$

□

con

```
\begin{proof}
Basta usare la formula
\[
E=mc^2.\qedhere
\]
\end{proof}
```

Dimostrazione. Basta usare la formula

$$E = mc^2.$$

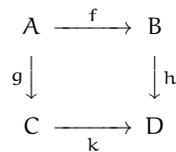
□

Come si può osservare, nel primo esempio il quadratino è collocato in modo (leggermente) sbagliato: la scrittura corretta è la seconda.

5.11 DIAGRAMMI COMMUTATIVI E DI VENN

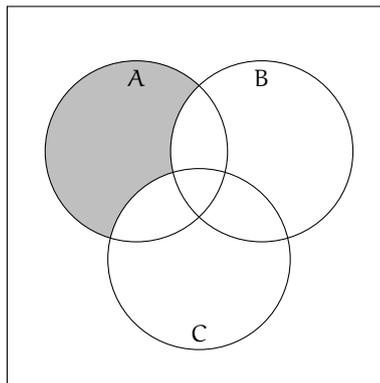
I diagrammi commutativi sono particolari oggetti a metà strada fra un insieme di espressioni matematiche e un disegno che le mette in relazione tra loro. Il pacchetto `amscd` definisce un ambiente `CD` (da usare solo in un altro ambiente per formule in `display`) con cui si possono disegnare diagrammi commutativi piani e senza frecce diagonali. Per diagrammi più complessi ci sono pacchetti più sofisticati, come `xypdf`.

```
\[
\begin{CD}
A @>f>> B \\
@V{g}VV @VV{h}V \\
C @>>k> D
\end{CD}
\]
```



Il pacchetto `venndiagram` permette di comporre semplici diagrammi di Venn.

```
\begin{venndiagram3sets}
\fillOnlyA
\end{venndiagram3sets}
```



5.12 ALTRE SCIENZE

Le unità di misura del Sistema Internazionale s'inseriscono con i comandi del pacchetto `siunitx` (se ne veda la documentazione), che permette di regolarne molto finemente il formato e di cambiare il risultato nel documento

6

TABELLE E FIGURE

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

6.1 STRUMENTI FONDAMENTALI

Per inserire tabelle e figure in un documento da comporre con $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ esistono tre strumenti essenzialmente:

- l’ambiente standard `tabular`, per tabelle che contengono prevalentemente testo;
- l’ambiente standard `array`, per tabelle che contengono prevalentemente matematica;
- il comando `\includegraphics` definito dal pacchetto `graphicx`, per includere nel documento le figure quando sono file *esterni* (come tutte quelle di questa guida).

Li si vede all’opera nei tre esempi seguenti.

```
La tabella
\begin{center}
\begin{tabular}{ll}
\toprule
Alcaloide & Origine \\
\midrule
atropina & belladonna \\
morfina & papavero \\
nicotina & tabacco \\
\bottomrule
\end{tabular}
\end{center}
mostra l’origine vegetale
di alcuni alcaloidi.
```

La tabella

Alcaloide	Origine
atropina	belladonna
morfina	papavero
nicotina	tabacco

mostra l’origine vegetale di alcuni alcaloidi.

```
La tabella
\[
\begin{array}{ll}
\toprule
f(x) & f'(x) \\
\midrule
x^n & nx^{n-1} \\
e^x & e^x \\
\sin x & \cos x \\
\bottomrule
\end{array}
\]
mostra le derivate
di alcune funzioni elementari.
```

La tabella

$f(x)$	$f'(x)$
x^n	nx^{n-1}
e^x	e^x
$\sin x$	$\cos x$

mostra le derivate di alcune funzioni elementari.

La figura
`\begin{center}`
`\includegraphics[width=%`
`0.5\textwidth]{Rettili}`
`\end{center}`
 riproduce
 l'incisione su legno
`\emph{Tassellazione`
 del piano con rettili}
 di M.-Escher.

La figura



riproduce l'incisione su legno *Tassellazione del piano con rettili* di M. Escher.

Si noti che:

- Tutti e tre *non* cominciano un nuovo capoverso, ma producono un'unità tipografica indivisibile che il programma tratta come se fosse un unico carattere, e che nel caso di `tabular` e `\includegraphics` va centrata rispetto alla giustezza del testo mettendola nell'ambiente `center`.
- Una tabella `array` va racchiusa a propria volta tra comandi matematici: se in testo, di solito si usano `\[...\]`, che sostituiscono l'ambiente `center` (ma se l'opzione di classe `fleqn` è attiva, la tabella non risulterà più centrata); se `mobile`, invece, si usano `$. .$.`.
- Quando richiesta, si consiglia di assegnare all'oggetto una larghezza *relativa* espressa con una frazione della giustezza stabilita dalla classe in uso (`\textwidth`, ma si veda anche il paragrafo 11.3.4 a pagina 179). Una larghezza *assoluta* causerebbe ovvi inconvenienti cambiando classe di documento o aumentando le colonne di composizione.
- L'ambiente `center` si omette anche quando si vuole in linea una figura particolarmente piccola, come mostra l'esempio seguente:

La mela morsicata
`\includegraphics[width=%`
`0.10\textwidth]{apple}`
 è il logo di Apple.

La mela morsicata  è il logo di Apple.

6.2 OGGETTI IN TESTO E FUORI TESTO

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

6.2.1 Tabelle e figure in testo

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

- appartengono al flusso del discorso e non possono esserne scorporati senza comprometterne la comprensione;
- non prevedono didascalia (proprio perché la loro funzione è spiegata nel contesto) né riferimenti incrociati a sé stessi;
- devono essere, perciò, quanto mai chiari e intuitivi.

Apparentemente innocui, oggetti di questo tipo possono comportare in realtà problemi di impaginazione a volte irrisolvibili. S'immagini, per esempio, di essere arrivati quasi alla fine della pagina, e di dover inserire *proprio*

li, perché richiesto dal discorso, una figura alta cinque centimetri avendone però soltanto tre a disposizione: va da sé che *li* la figura *non* ci può stare in nessun modo: se lo spazio fisico non c'è, non lo si può inventare! Questa situazione, si badi bene, non si verifica solo con \LaTeX , ma si dà indipendentemente dal programma in uso per scrivere. Come fare?

Una prima soluzione ingenua (e da evitare) potrebbe essere quella di cominciare una nuova pagina ogni volta che un oggetto non può stare in quella corrente; in questo modo, però, le pagine interrotte rimarrebbero parzialmente bianche, con un risultato tipografico insoddisfacente.

Ecco perché gli oggetti in testo devono essere *eccezionali* (tornano utili per mettere un logo “proprio *li*” e in pochissime altre circostanze) e di piccole o piccolissime dimensioni.

6.2.2 Tabelle e figure fuori testo

Si risolve il problema rendendo gli oggetti “fuori testo” (o “mobili”; in inglese *floating*, “galleggianti”) e lasciando fare a \LaTeX che, nell'esatto ordine in cui oggetti dello stesso tipo sono definiti nel sorgente, li metterà nel punto *per lui* migliore (sulla pagina corrente se ci stanno, oppure in pagine *successive* a quella in cui finirebbero) riempiendo lo spazio rimanente con l'altro materiale a disposizione. Questa soluzione è molto vantaggiosa, perché garantisce l'ottimale riempimento della pagina tipico di \LaTeX .

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

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

Indipendentemente dalle dimensioni del documento e degli oggetti, perciò, si raccomanda di includerli *sempre* così (come si è fatto in tutta questa guida), vincendo quanto prima le iniziali e comprensibili perplessità derivanti dal vederli molto spesso in un punto diverso da quello in cui li si è definiti, come si spiegherà tra poco.

Nei prossimi paragrafi si spiegano gli ambienti mobili e come comportarsi durante la stesura del documento; nel paragrafo 12.2 a pagina 186 si mostra che cosa (eventualmente) si può fare durante la revisione per risolvere collocazioni poco gradite e migliorarlo ulteriormente.

Gli ambienti table e figure

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

```
\begin{table}[<preferenze di collocazione>]
...
\end{table}
```

se è una tabella, oppure in quello `figure`

```
\begin{figure}[<preferenze di collocazione>]
...
\end{figure}
```

se è una figura. Come si può notare, entrambi accettano l'argomento facoltativo *<preferenze di collocazione>*, il cui funzionamento verrà spiegato nel paragrafo 6.2.2 a pagina 96. (Se si sta componendo un documento a due colonne

si possono usare le varianti asterisco dei due ambienti, che mettono l'oggetto sull'intera pagina e non sulla singola colonna di composizione.) I due ambienti appena esaminati richiedono alcuni comandi importanti, descritti di seguito.

Il comando

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

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

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

che produce didascalie prive di intestazione e numero, utili per esempio in tavole illustrate fuori testo o in un libro d'arte.

I due comandi

```
\listoftables
\listoffigures
```

producono *nel punto in cui li si dà* rispettivamente la sezione contenente l'elenco delle tabelle o delle figure (esattamente come fa `\tableofcontents` per l'indice generale) con relativi titolo e testatina. Le loro voci sono le stesse didascalie degli oggetti o, se troppo lunghe, una versione ridotta da scrivere nell'argomento facoltativo di `\caption`:

```
\caption[<didascalia breve>]{<didascalia normale>}
```

Si badi bene a mettere i due indici appena visti solo se il numero degli oggetti è rilevante o è importante per il lettore poterli ritrovare agevolmente: un elenco di sole due figure, per esempio, non avrebbe alcuna utilità. Anche questi due indici richiedono *due* composizioni successive.

Il comando `\label`, da dare sempre *dopo* il corrispondente `\caption`, assegna all'oggetto un'etichetta per i riferimenti incrociati (si veda il paragrafo 3.10 a pagina 39).

Codici tipo

Il modo migliore per introdurre un oggetto mobile nel sorgente è scriverne il relativo ambiente preceduto e seguito da una riga vuota. Ecco un esempio tipico per una tabella `tabular` (in modo del tutto analogo si compone una tabella `array`, ricordandosi di racchiuderla tra dollari) con il relativo richiamo:

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

```
\begin{table}
\caption{(..)}
\label{tab:esempio}
\centering
\begin{tabular}{(..)}
...
\end{tabular}
\end{table}
```

La tabella-`\ref{tab:esempio}` è un esempio di tabella mobile.

E uno per una figura:

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

```
\begin{figure}
```

```

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

```

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

Come si può osservare:

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

Le classi standard prevedono la didascalia sempre *sotto* l'oggetto, perciò \LaTeX non la "riconosce" se messa sopra e la restituisce in modo non appropriato mettendogliela troppo a ridosso. Risolve il problema il pacchetto `caption`, da impostare come segue:

```

\usepackage{caption}
\captionsetup{tableposition=top,figureposition=bottom,font=small}

```

dove:

- `tableposition=top` ordina al programma di inserire uno spazio adeguato tra didascalia e tabella;
- `figureposition=bottom` è l'opzione analoga alla precedente per le figure;
- `font=small` produce didascalie in corpo più piccolo, secondo l'uso italiano.

Quando le opzioni sono numerose o la loro forma lo richiede, si possono scrivere con il comando `\captionsetup`.

Si noti che molte classi non standard modificano il rapporto tra ambiente mobile e didascalia per questioni di stile. Il risultato finale, perciò, *potrebbe* non corrispondere a quello appena descritto: si faccia qualche prova. Infine, si tenga presente che `caption` aumenta lo spazio tra didascalia e oggetto anche solo caricandolo.

Personalizzare la didascalia

Il pacchetto `caption` permette anche di personalizzare finemente le didascalie in *ogni* loro aspetto. Quelle di questa guida, per esempio, sono state composte aggiungendo le seguenti opzioni a quelle appena viste:

```

\captionsetup{format=hang,labelfont={sf,bf}}

```

dove:

- `format=hang` allinea (*hang*) alla prima riga quelle successive (\LaTeX centra automaticamente le didascalie che occupano una sola riga);
- `labelfont={sf,bf}` imposta l'etichetta della didascalia in caratteri senza grazie neri.

Tabella 30: Preferenze di collocazione per gli oggetti mobili.

Preferenza	Chiede a \LaTeX di mettere l'oggetto
h	Qui (<i>here</i>), se possibile
t	In cima (<i>top</i>) alla pagina
b	In fondo (<i>bottom</i>) alla pagina
p	In una pagina di soli oggetti mobili (<i>page of floats</i>)
!	Dove vorrebbe l'utente per quanto possibile

Che cosa fare durante la stesura

Durante la stesura del documento, si sa, bisogna concentrarsi sul contenuto del lavoro, lasciando fare a \LaTeX tutto il resto. La gestione degli oggetti non si sottrae a questa regola, per cui in prima battuta si consiglia di inserirli tutti *senza specificare alcuna preferenza di collocazione* (si veda poco sotto): il risultato è generalmente ottimo. In linea di massima si troveranno gli oggetti “abbastanza” vicini al punto in cui stanno nel sorgente, e in particolare:

- quasi sempre in cima alla pagina (corrente o una delle successive);
- raramente in basso;
- se sono grandi (secondo i parametri di \LaTeX), in un pagina di soli oggetti mobili, sempre successiva al punto in cui li si è messi nel sorgente.

Basta sfogliare un qualunque libro ben composto per verificare che le cose stanno proprio così.

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

I risultati automatici sono *quasi sempre* migliori di quelli che si potrebbero ottenere cercando di collocare gli oggetti a mano. Quando i gusti dell'utente non coincidono con quelli del programma, tuttavia, si possono usare le *preferenze di collocazione* raccolte nella tabella 30, che “suggeriscono” a \LaTeX come si vorrebbero vedere gli oggetti sulle pagine del proprio documento. \LaTeX seguirà i suggerimenti nell'ordine in cui li trova (fa eccezione la preferenza p, come si spiega nel paragrafo 12.2 a pagina 186). Di seguito si propongono un paio delle possibilità più usate:

- tp se non si vuole nessun oggetto in fondo alla pagina;
- htp se si vuole che \LaTeX cerchi come prima cosa di mettere l'oggetto esattamente lì dove lo si è inserito (se è sufficientemente piccolo, di solito si viene accontentati).

Ecco ora le opzioni da evitare *sempre*:

- h o, peggio, h! *possono* funzionare solo con oggetti molto piccoli; in caso contrario, l'oggetto viene messo alla fine del capitolo (o del documento) portandosi dietro tutti gli altri inseriti successivamente (si tenga *ben* presente questo comportamento);
- t e b da sole, perché è buona regola dare al programma almeno un paio di possibilità (ma è ammessa la sola p).

In casi estremi, e solo per ottenere effetti particolari, si può forzare l'oggetto nella posizione desiderata con la preferenza H del pacchetto float (da usare *sempre* da sola).

Tabella 31: Tabella che non rispetta le regole generali.

D	P	u	β	G
.500 m	269.8 kg	.000674 m	1.79	.04089 Pa
1.50 m	421.0 kg	.001035 m	3.59	"
10.0 m	640.2 kg	.001565 m	7.18	"

Tabella 32: Tabella che rispetta le regole generali.

D (m)	P (kg)	u (m)	β	G (Pa)
0,500	269,8	0,000 674	1,79	0,040 89
1,50	421,0	0,001 035	3,59	0,040 89
10,0	640,2	0,001 565	7,18	0,040 89

```
\dots
```

qui finisce un capoverso.

```
\begin{figure}[H]
\centering
\includegraphics[width=0.5\textwidth]{Formica}
\caption{Figura collocata a mano.}
\label{fig:float}
\end{figure}
```

La figura~\vref{fig:float} è un esempio di figura mobile collocata a mano.

... qui finisce un capoverso.

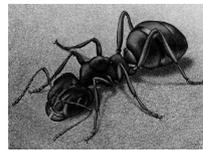


Figura 7: Figura collocata a mano.

La figura 7 è un esempio di figura mobile collocata a mano.

6.3 TABELLE

Comporre tabelle di altissima qualità è una delle specialità di \LaTeX , ma i suoi comandi standard sono piuttosto limitati. Numerosi pacchetti, però, ne definiscono di nuovi e più avanzati con cui si può personalizzare finemente il proprio lavoro. Questo paragrafo, basato su [Mori, 2006], cui si rimanda per gli approfondimenti, spiega come usare gli uni e gli altri e affronta gli aspetti principali dell'argomento.

6.3.1 Indicazioni generali

Regole generali di composizione

Le regole seguenti permettono di ottenere una tabella ben composta:

- non usare *mai* filetti verticali;
- evitare filetti doppi;
- scrivere *sempre* le unità di misura nell'intestazione della colonna e non nel corpo della tabella;
- non usare *mai* le virgolette per ripetere il contenuto di una cella;
- far precedere *sempre* il separatore decimale da almeno una cifra.

A queste si possono aggiungere le seguenti:

- incolonnare i numeri al separatore decimale, se presente, che nei documenti in italiano è la virgola;
- usare i numeri maiuscoli nelle tabelle numeriche (si veda il paragrafo A.4.2 a pagina 195).

Così vuole la tradizione tipografica, in contrasto con la cattiva abitudine, purtroppo oggi molto diffusa, di comporre le tabelle come se fossero parti di un foglio elettronico. Per capire quanto sia importante rispettare queste regole, si confrontino le tabelle 31 e 32 nella pagina precedente.

Separare le celle e chiudere le righe

Le celle di una tabella vanno separate tra loro con il carattere separatore & e le righe *devono* terminare con il comando `\\`, pena un errore. Si noti che se una riga ha meno celle piene di quante sono le colonne, può essere chiusa dopo l'ultima cella riempita.

Filetti professionali: booktabs

Filetti migliori di quelli che si ottengono con il comando standard `\hline`, dalla resa tipografica insoddisfacente per via dello spazio troppo risicato che risulta tra filetti e testo nelle celle, sono prodotti da tre comandi definiti dal pacchetto `booktabs`. Questi comandi non vogliono `\\` dopo di sé, producono filetti di spessore differente (le righe prodotte da `\midrule`, infatti, sono più sottili delle altre) e vanno dati secondo un ordine rigoroso:

1. `\toprule` produce il primo filetto della tabella;
2. `\midrule` produce il filetto interno (o, ripetendolo, i filetti, ma non se ne abusi);
3. `\bottomrule` produce l'ultimo filetto.

Il comando

```
\cmidrule{<n>-<m>}
```

invece, disegna un filetto orizzontale dalla sinistra della colonna $\langle n \rangle$ -esima fino alla destra della colonna $\langle m \rangle$ -esima. Per migliorare leggermente la resa della tabella, si può specificare subito dopo `\cmidrule` un argomento facoltativo $\langle \text{troncamento} \rangle$ fra parentesi tonde:

```
\cmidrule(<troncamento>){<n>-<m>}
```

che accetta tre possibilità: `r`, `l` o `rl`. Si sta dicendo a \LaTeX che il filetto va "rasato" a destra (`r`), a sinistra (`l`) o a entrambe le estremità (`rl`).

Si possono rasare anche i filetti prodotti dai primi tre comandi visti semplicemente scrivendo `@{}` all'inizio e alla fine del preambolo della tabella (si veda il paragrafo 6.3.2 a fronte): potrebbe servire se lo spazio è poco o semplicemente come tocco di stile.

Codice ordinato

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

Tabella 33: Descrittori standard delle colonne.

Descrittore	Spiegazione
l	Allinea il contenuto della cella a sinistra (<i>left</i>)
c	Centra il contenuto della cella (<i>center</i>)
r	Allinea il contenuto della cella a destra (<i>right</i>)
p	Giustifica un testo lungo entro una <i><larghezza></i>
*	Ripete i descrittori

6.3.2 Tabelle standard: `tabular` e `array`

In linea generale, una tabella che contiene prevalentemente testo va composta dentro l'ambiente `tabular`; una tabella che contiene prevalentemente matematica va composta dentro l'ambiente `array`. Entrambi si comportano in modo molto simile, come si può osservare negli esempi seguenti:

```
\begin{tabular}{lcr}
\toprule
Grandezza & Simbolo & Unità \\
\midrule
forza & $F$ & newton \\
energia & $E$ & joule \\
tensione & $V$ & volt \\
\bottomrule
\end{tabular}
```

Grandezza	Simbolo	Unità
forza	F	newton
energia	E	joule
tensione	V	volt

```
\[
\begin{array}{cc}
\toprule
f(x) & \text{Una primitiva} \\
\midrule
e^x & e^x \\
\cos x & \sin x \\
\sin x & -\cos x \\
\bottomrule
\end{array}
\]
```

f(x)	Una primitiva
e^x	e^x
$\cos x$	$\sin x$
$\sin x$	$-\cos x$

Alle osservazioni del paragrafo 6.1 a pagina 91 si aggiungano le seguenti:

- `tabular` e `array` richiedono un argomento, detto *preambolo della tabella*, formato da un certo numero di *descrittori*, ciascuno dei quali definisce il comportamento di un *tipo di colonna* come spiegato nella tabella 33;
- in `tabular`, eventuali formule matematiche si scrivono con i comandi per le formule in linea, per esempio fra dollari $\$. . \$$;
- in `array`, un eventuale testo si scrive nell'argomento del comando `\text` del pacchetto `amsmath`, che va dunque caricato.

Si noti che per "alleggerire" il codice, nel preambolo di una tabella si può *sempre* usare la scrittura

```
*{<n>}{<descrittore>}
```

equivalente a $\langle n \rangle$ colonne consecutive del tipo indicato dal $\langle \text{descrittore} \rangle$.

Tabella 34: Tabella con colonna p.

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

6.3.3 Celle con testo troppo lungo

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

- a un descrittore `p{larghezza}`, che permette di stabilire a priori la larghezza di una sola colonna;
- al pacchetto `tabularx`, che permette di stabilire a priori la larghezza dell'intera tabella.

Si noti che in entrambi i casi le eventuali intestazioni di colonna vengono allineate a sinistra per impostazione predefinita: per cambiare questo risultato si usi il comando `\multicolumn` spiegato nel paragrafo 6.3.5 a pagina 104.

Colonne di larghezza prefissata

Il codice seguente, che mostra all'opera il descrittore `p{larghezza}`, produce la tabella 34:

```
\begin{tabular}{lp{0.5\textwidth}}
\toprule
\textbf{Forza} & Una forza è una grandezza fisica che si manifesta
nell'interazione di due o più corpi materiali, che cambia lo stato
di quiete o di moto dei corpi stessi. \\
\midrule
\textbf{Momento polare} & Il momento polare di una forza rispetto a
una determinata origine è definito come il prodotto vettoriale tra
il vettore posizione (rispetto alla stessa origine) e la forza. \\
\bottomrule
\end{tabular}
```

Si noti che:

- `\textbf` produce il proprio argomento in nero;
- per impostazione predefinita, il contenuto di una colonna p viene giustificato e sillabato automaticamente;
- le eventuali colonne `l`, `c` e `r` rimangono della propria larghezza naturale;
- la cella è allineata alla riga a cui appartiene rispetto alla linea di base superiore.

Tabella 35: Tabella di larghezza prefissata ottenuta con tabularx.

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

Tabella 36: Tabella con due colonne della della stessa larghezza ottenuta con tabularx.

Periodo	Fenomeni geologici	Biosfera
Giurassico	Periodo caratterizzato da variazioni del livello del mare; prevalenza delle terre emerse in America, Asia, Australia.	Fauna: compaiono i primi marsupiali; dominano i grandi rettili (dinosauri). Flora: predominano le conifere.
Triassico	Intensa l'erosione dei continenti; profonde fratture da cui escono lave che originano altipiani estesi.	Fauna: si diffondono i rettili; nei mari prosperano pesci e invertebrati. Flora: si sviluppano alghe caratteristiche.

Tabelle di larghezza prefissata

Il pacchetto `tabularx`, che carica il pacchetto `array` (si veda il paragrafo 6.3.8 a pagina 106), definisce l'omonimo ambiente `tabularx` e un nuovo tipo di colonna X , che alle caratteristiche delle colonne p appena esaminate aggiunge un vantaggio: è \LaTeX a calcolarne automaticamente la larghezza in base alla larghezza complessiva assegnata *all'intera tabella*. Infatti `tabularx` richiede *obbligatoriamente* un secondo argomento in cui indicarla (nei prossimi esempi è pari a `\textwidth`, ma sono ammessi anche altri valori, da impostare come già spiegato nel paragrafo 6.1 a pagina 91). Si osserva il pacchetto all'opera nella tabella 35, identica alla 34 nel contenuto, ottenuta con il codice seguente:

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

La tabella 36, prodotta con il codice seguente, mostra come a *tutte* le colonne X , se più d'una, \LaTeX assegni la stessa larghezza *indipendentemente* dalle altre colonne presenti.

```
\begin{tabularx}{\textwidth}{lXX}
\toprule
Periodo & Fenomeni geologici & Biosfera \\
\midrule
\textbf{Giurassico} & Periodo caratterizzato da variazioni del
livello del mare; prevalenza delle terre emerse in America, Asia,
```

Tabella 37: Tabella con colonna S.

Espressione	Valore
π	3,1416
π^π	36,46
π^{π^π}	80 662,7

```
Australia. & Fauna: compaiono i primi marsupiali; dominano i grandi
rettili (dinosauri). Flora: predominano le conifere. \\
\midrule
\textbf{Triassico} & Intensa l'erosione dei continenti; profonde
fratture da cui escono lave che originano altopiani estesi.
& Fauna: si diffondono i rettili; nei mari prosperano pesci e
invertebrati. Flora: si sviluppano alghe caratteristiche. \\
\bottomrule
\end{tabularx}
```

6.3.4 Colonne di soli numeri: siunitx

Il pacchetto `siunitx` gestisce in modo molto potente e flessibile anche la resa tipografica dei numeri nelle tabelle, definendo un nuovo tipo di colonna *S* *specificata per dati numerici* che si comporta come segue:

- nei numeri di cinque o più cifre separa automaticamente le cifre a gruppi di tre per migliorarne la leggibilità (si veda il paragrafo [A.4.2](#) a pagina [195](#));
- per impostazione predefinita, mette il separatore decimale al centro della colonna, di cui espande flessibilmente entrambi i margini in base alla lunghezza del numero;
- spazia correttamente il separatore decimale (se c'è) tra parte intera e decimale del numero (si veda il paragrafo [A.4.2](#) a pagina [195](#)).

Lo si vede all'opera nella tabella [37](#), ottenuta con il seguente codice:

```
\usepackage{siunitx}
\sisetup{output-decimal-marker={,}}

\begin{tabular}{cS}
\toprule
Espressione & {Valore} \\
\midrule
 $\pi$  & 3.1416 \\
 $\pi^\pi$  & 36.46 \\
 $\pi^{\pi^\pi}$  & 80662.7 \\
\bottomrule
\end{tabular}
```

Si noti che:

- si possono scrivere le opzioni di `siunitx` (specie se numerose) anche con il comando `\sisetup` immediatamente dopo aver caricato il pacchetto;
- dell'eventuale testo in una colonna *S* (di solito nell'intestazione) si scrive tra parentesi graffe per non "confondere" il pacchetto, che lì si aspetterebbe dei numeri.

Tabella 38: Tabella con cella multicolonna.

Particella	Nome		Carica (e)
	Particella	Antiparticella	
elettrone	positrone		∓ 1
protone	antiprotone		± 1
neutrone	antineutrone		0

Tra le numerose opzioni che il pacchetto definisce per trattare i numeri nelle tabelle, se ne segnalano qui un paio, utili per risolvere alcune situazioni in cui il comportamento predefinito delle colonne S risulta inadeguato. Nell'esempio seguente si nota uno spazio tra le colonne decisamente esagerato:

P (kg)	u (m)
269,8	0,000 674
421,0	0,001 035
640,2	0,001 565

Risolve il problema l'opzione `table-format=<valore>`, da assegnare a ciascuna colonna S nel preambolo della tabella. La sintassi completa dell'opzione è:

```
table-format=<prima>.<dopo>
```

dove:

- *<prima>* è il numero di cifre intere del numero che nella colonna ha la parte intera più lunga;
- *'.'* è il separatore decimale, che nel documento finito sarà reso con la virgola;
- *<dopo>* è il numero di cifre decimali del numero che nella colonna ha la parte decimale più lunga.

La tabella precedente diventa:

P (kg)	u (m)
269,8	0,000 674
421,0	0,001 035
640,2	0,001 565

Nello stesso modo è stata prodotta la tabella 32 a pagina 97.

Quando c'è poco spazio nella tabella ma non s'intende rinunciare alle colonne S, risolve il problema l'opzione `table-parse-only`, che in pratica si comporta come un descrittore standard c. Eccola all'opera:

Tabella 39: Tabella con celle multiriga.

Famiglia	Particella	Simbolo
leptoni	elettrone	e
	muone	μ
	tau	τ
	neutrino	ν
barioni	protone	p
	neutrone	n

```

\begin{tabular}%
  {SS[table-parse-only]}
\toprule
{Allineati} & {Centrati} \\
\midrule
3.1416      & 3.1416 \\
36.46      & 36.46 \\
80662.7    & 80662.7 \\
\bottomrule
\end{tabular}

```

	Allineati	Centrati
	3,1416	3,1416
	36,46	36,46
	80 662,7	80 662,7

Se in un documento le tabelle numeriche sono poche o pochissime, siunitx potrebbe rivelarsi troppo laborioso: si abbia cura allora di allineare i numeri a destra e di assegnare loro lo stesso numero di cifre decimali.

6.3.5 Celle multicolonna

Il comando

```
\multicolumn{<n>}{<descrittore>}{<testo>}
```

sostituisce a $\langle n \rangle$ celle successive un'unica cella, il cui $\langle testo \rangle$ viene organizzato nei modi specificati con il $\langle descrittore \rangle$. Lo si vede all'opera nel prossimo esempio, che produce la tabella 38 nella pagina precedente:

```

\begin{tabular}{llc}
\toprule
\multicolumn{2}{c}{Nome} & Carica \\
Particella & Antiparticella & (e) \\
\midrule
elettrone & positrone &  $\mp 1$  \\
protone & antiprotone &  $\pm 1$  \\
neutrone & antineutrone &  $0$  \\
\bottomrule
\end{tabular}

```

Si può usare il comando anche per *una sola* cella, come spesso accade nell'intestazione della tabella: in questo caso, a $\langle n \rangle$ si sostituisce 1.

6.3.6 Celle multiriga

Il comando

```
\multirow{<n>}{<testo>}
```

(che richiede il pacchetto omonimo) crea una cella alta $\langle n \rangle$ righe, il cui $\langle testo \rangle$ verrà centrato verticalmente. Il codice

Tabella 40: Tabella con celle multiriga e multicolonna.

Elemento	Strati		
	K	L	M
idrogeno	1		
litio	2	1	
sodio	2	8	1

```

\begin{tabular}{clc}
\toprule
Famiglia & Particella & Simbolo \\
\midrule
\multirow{4}{*}{leptoni} & elettrone &  $e^-$  \\
& muone &  $\mu^-$  \\
& tau &  $\tau^-$  \\
& neutrino &  $\nu$  \\
\midrule
\multirow{2}{*}{barioni} & protone &  $p$  \\
& neutrone &  $n$  \\
\bottomrule
\end{tabular}

```

produce la tabella 39 a fronte. Si noti, però, che le celle multiriga:

- *non sempre* saranno perfettamente centrate rispetto alle righe vicine;
- sono incompatibili con le colonne X.

I due comandi si possono combinare. Il codice

```

\begin{tabular}{lccc}
\toprule
\multirow{2}{*}{Elemento} & \multicolumn{3}{c}{Strati} \\
\cmidrule(lr){2-4}
& K & L & M \\
\midrule
idrogeno & 1 & & \\
litio & 2 & 1 & \\
sodio & 2 & 8 & 1 \\
\bottomrule
\end{tabular}

```

produce la tabella 40.

6.3.7 Spaziare a mano righe e colonne

Quando (molto di rado) i risultati di \LaTeX non soddisfano completamente, si può migliorare la resa tipografica della tabella con piccoli aggiustamenti manuali. I casi, di solito, sono due:

1. le righe della tabella appaiono troppo ravvicinate;
2. le colonne della tabella appaiono troppo ravvicinate.

Risolve il primo caso il comando

```
\[{\altezza}]
```

che, sostituito al delimitatore di riga standard `\` nel corpo della tabella, abbassa la riga immediatamente successiva (e solo quella) di uno spazio verticale pari ad $\langle \text{altezza} \rangle$. Di solito s’inserisce questo spazio supplementare per rimediare a lievi sovrapposizioni (specialmente di formule matematiche in display) oppure per dare maggiore “respiro” alla tabella come si è fatto nella tabella 44 a pagina 111, dove si sono allontanati leggermente i blocchi di testo altrimenti troppo vicini.

Si osservi come la spaziatura automatica nella tabella seguente è troppo risicata

<pre>\[\begin{array}{cc} \toprule f(x) & f'(x) \\ \midrule \log x & \dfrac{1}{x} \\ \arctan x & \dfrac{1}{1+x^2} \\ \bottomrule \end{array} \]</pre>	<table style="margin: auto; border-collapse: collapse;"> <tr> <td style="border-top: 1px solid black; border-bottom: 1px solid black; padding: 5px;">f(x)</td> <td style="border-top: 1px solid black; border-bottom: 1px solid black; padding: 5px;">f'(x)</td> </tr> <tr> <td style="padding: 5px;">log x</td> <td style="padding: 5px;">$\frac{1}{x}$</td> </tr> <tr> <td style="padding: 5px;">arctan x</td> <td style="padding: 5px;">$\frac{1}{1+x^2}$</td> </tr> </table>	f(x)	f'(x)	log x	$\frac{1}{x}$	arctan x	$\frac{1}{1+x^2}$
f(x)	f'(x)						
log x	$\frac{1}{x}$						
arctan x	$\frac{1}{1+x^2}$						

e come la si può correggere molto semplicemente con

<pre>\[\begin{array}{cc} \toprule f(x) & f'(x) \\ \midrule \log x & \dfrac{1}{x} \\ \arctan x & \dfrac{1}{1+x^2} \\ \bottomrule \end{array} \]</pre>	<table style="margin: auto; border-collapse: collapse;"> <tr> <td style="border-top: 1px solid black; border-bottom: 1px solid black; padding: 5px;">f(x)</td> <td style="border-top: 1px solid black; border-bottom: 1px solid black; padding: 5px;">f'(x)</td> </tr> <tr> <td style="padding: 5px;">log x</td> <td style="padding: 5px;">$\frac{1}{x}$</td> </tr> <tr> <td style="padding: 5px;">arctan x</td> <td style="padding: 5px;">$\frac{1}{1+x^2}$</td> </tr> </table>	f(x)	f'(x)	log x	$\frac{1}{x}$	arctan x	$\frac{1}{1+x^2}$
f(x)	f'(x)						
log x	$\frac{1}{x}$						
arctan x	$\frac{1}{1+x^2}$						

Il comando `\dfrac` richiede il pacchetto `amsmath`.

Non accade praticamente *mai* di dover aumentare la distanza tra due colonne. Se proprio ce ne fosse bisogno, si può usare il separatore

`@{\langle larghezza \rangle}`

che, dato nel *preambolo della tabella* tra i descrittori delle due colonne che si desidera allontanare, inserisce uno spazio orizzontale pari a $\langle \text{larghezza} \rangle$. Si noti che *in questo caso* la $\langle \text{larghezza} \rangle$ va indicata:

- mettendo le consuete espressioni (valore e unità di misura) nell’argomento di `\hspace{\langle \dots \rangle}`;
- con comandi di spaziatura fissa (`\quad` e `\qquad`, per esempio).

Le due istruzioni appena esaminate si possono eseguire insieme e anche più di una volta nella stessa tabella, là dove serve.

6.3.8 Personalizzare le colonne: array

I diversi tipi di colonna esaminati fino a qui (gli standard `l`, `c`, `r` e `p` e gli “speciali” `X` e `S`) riescono a soddisfare la maggior parte delle esigenze, ma non si possono personalizzare. S’immagini, per esempio, di dover scrivere il contenuto di un’intera colonna in nero: può essere più vantaggioso, soprattutto se le celle sono numerose, definire questo stile una volta per tutte anziché farlo in ognuna.

Risolve il problema il *pacchetto* array (da non confondere con l'omonimo ambiente standard), che per personalizzare le colonne definisce le due istruzioni $\>\{ \langle \text{dichiarazione iniziale} \rangle \}$ e $\langle \{ \langle \text{dichiarazione finale} \rangle \}$, la cui sintassi completa è

```
 $\>\{ \langle \text{dichiarazione iniziale} \rangle \} \langle \text{descrittore} \rangle \langle \{ \langle \text{dichiarazione finale} \rangle \}$ 
```

dove:

- $\>\{ \langle \text{dichiarazione iniziale} \rangle \}$ va data prima del $\langle \text{descrittore} \rangle$ per dire a L^AT_EX di eseguirla *prima* del contenuto della colonna;
- $\langle \text{descrittore} \rangle$ è un descrittore di colonna, di quelli già esaminati o eventualmente definito con `\newcolumnmtype` (si veda più sotto);
- $\langle \{ \langle \text{dichiarazione finale} \rangle \}$ va data dopo il $\langle \text{descrittore} \rangle$ per dire a L^AT_EX di eseguirla *dopo* il contenuto della colonna.

Le istruzioni appena viste si possono usare nel preambolo *della tabella* se servono ogni tanto; se invece le tabelle con colonne personalizzate sono molte, è più conveniente definirle una volta per tutte nel preambolo *del documento* con il comando `\newcolumnmtype`, la cui sintassi completa è

```
\newcolumnmtype{ \langle carattere \rangle }{ \>\{ \langle dichiarazione \rangle \} \langle descrittore \rangle \langle \{ \langle dichiarazione \rangle \} }
```

dove $\langle \text{carattere} \rangle$ è una lettera, diversa da tutti i descrittori di colonna già in uso nel documento, che identifica il nuovo descrittore, da usare come al solito nel preambolo della tabella.

Nei prossimi paragrafi si vedono le dichiarazioni all'opera.

Modificare lo stile del font di una colonna

Le caratteristiche di array tornano utili per modificare lo stile del font di una colonna: basta sostituire a $\langle \text{dichiarazione iniziale} \rangle$ una delle dichiarazioni elencate nella tabella 18 a pagina 55. Il codice

```
\begin{tabular}{>{\bfseries}lp{0.5\textwidth}}
\toprule
Forza & Una forza è una grandezza fisica che si manifesta
nell'interazione di due o più corpi materiali, che cambia lo stato
di quiete o di moto dei corpi stessi. \\
\midrule
Momento polare & Il momento polare di una forza rispetto a una
determinata origine è definito come il prodotto vettoriale tra il
vettore posizione (rispetto alla stessa origine) e la forza. \\
\bottomrule
\end{tabular}
```

produce la tabella 34 a pagina 100, prodotta in quella sede specificando a mano lo stile della prima colonna.

Modificare l'allineamento del testo in una colonna

Le istruzioni appena viste possono servire anche quando una colonna p o X fosse troppo stretta per giustificare il testo in modo soddisfacente. Basta sostituire a $\langle \text{dichiarazione iniziale} \rangle$ una delle tre dichiarazioni seguenti:

- `\raggedright` (“disordinato a destra”) per avere il contenuto delle celle allineato *a sinistra*;
- `\centering` per averlo centrato;
- `\raggedleft` (“disordinato a sinistra”) per averlo allineato *a destra*.

Tabella 41: Tabelle con testo diversamente allineato in una colonna p.

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

Si confrontino le tabelle 41a e 41b, la seconda delle quali è stata ottenuta con il codice seguente:

```
\begin{tabular}{>{\bfseries}l>{\raggedright\arraybackslash}%
                p{0.2\columnwidth}}
\toprule
Grandezza & Descrizione \\
\midrule
Forza & Una forza è una grandezza fisica che si manifesta
nell'interazione di due o più corpi materiali, che cambia lo
stato di quiete o di moto dei corpi stessi. \\
\midrule
Momento polare & Il momento polare di una forza rispetto a
una determinata origine è definito come il prodotto vettoriale tra
il vettore posizione (rispetto alla stessa origine) e la forza. \\
\bottomrule
\end{tabular}
```

Si noti che:

- Diversamente da quanto accade in una colonna p standard, l'intestazione di colonna segue l'allineamento dichiarato.
- Se *nell'ultima colonna* della tabella c'è una delle tre dichiarazioni appena viste, la *riga della tabella* va terminata con `\tabularnewline`, mentre si userà `\\` per andare a capo *nella stessa cella*. Si può usare il consueto `\\` solo dando anche `\arraybackslash` come mostrato.
- Il testo *non è sillabato*. Per sillabarlo, basta caricare l'apposito pacchetto `ragged2e` e sostituire a quelli usati i comandi `\RaggedRight` e `\RaggedLeft`.

Una colonna X si comporta in modo analogo.

Colonne di sola matematica

I comandi seguenti

```
\newcolumnntype{L}{>{$}l<{$}}
\newcolumnntype{C}{>{$}c<{$}}
\newcolumnntype{R}{>{$}r<{$}}
```

Tabella 42: Tabella ottenuta con array.

$\int \cos x \, dx$	$\sin x + c$
$\int e^x \, dx$	$e^x + c$
$\int \sec^2 x \, dx$	$\tan x + c$

definiscono tre nuove colonne L, C e R nelle quali le formule matematiche vengono rispettivamente allineate a sinistra, centrate e allineate a destra. Se si preferisce il formato in display, basta aggiungere il relativo comando alla definizione della nuova colonna. Il codice

```
\newcolumnntype{L}{>{\displaystyle}l<{}}
\newcolumnntype{C}{>{\displaystyle}c<{}}

\begin{tabular}{LC}
\toprule
\int \cos x \, dx & \sin x + c \\
\midrule
\int e^x \, dx & e^x + c \\
\midrule
\int \sec^2 x \, dx & \tan x + c \\
\bottomrule
\end{tabular}
```

produce la tabella 42.

Ulteriori allineamenti

Il pacchetto array definisce ulteriori due descrittori, m e b. Con entrambi il contenuto delle celle è composto come in una colonna p, ma:

- con m la cella è centrata rispetto alla riga a cui appartiene;
- con b la cella è allineata rispetto alla riga a cui appartiene rispetto alla linea di base inferiore.

Per ulteriori dettagli su array si consiglia la lettura della sua documentazione e di [Gregorio, 2010].

6.3.9 Tabelle con note

Il comando `\footnote`, che in un testo produce le note al piede, non funziona nell'ambiente `tabular` e, come per la maggior parte delle cose in \LaTeX , questa limitazione ha ottime ragioni per esserci. Come si sa, una tabella dovrebbe essere *sempre* inserita fuori testo in un documento: una nota al piede la vincolerebbe alla pagina in cui si trova la nota.

Il luogo più adatto per mettere eventuali annotazioni è la didascalia, ma a volte le note esplicative nella tabella sono imposte da relatore o editore. Risolve il problema il pacchetto `threeparttable`, semplicissimo da usare, con il quale si è prodotta la tabella 11 a pagina 38.

6.3.10 Tabelle grandi

Se le dimensioni della tabella finita eccedono quelle della gabbia del testo in lunghezza, in larghezza o in entrambe, si prospettano soluzioni diverse

Tabella 43: Tabella con corpo di carattere inferiore (`footnotesize`) a quello del testo principale (`normalsize`).

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

a seconda della dimensione in eccesso. Se *troppo lunga* o *troppo larga*, si può spezzarla su più pagine, ridurre il corpo del font o ruotarla (ma quest'ultima soluzione si applica solo se troppo larga). A ogni caso si possono applicare più soluzioni contemporaneamente.

Ridurre il corpo del font

Per ridurre il corpo del font in una tabella (ma non nell'eventuale didascalia) si usano le stesse dichiarazioni elencate nella tabella 19 a pagina 56, con l'avvertenza di darle:

- *fuori* da `tabular` se la tabella è *in testo* (subito dopo la tabella una nuova dichiarazione ripristinerà il font corrente);
- *dentro* `table` se la tabella è *mobile* (si veda il paragrafo 6.2 a pagina 92).

Il codice

```
\begin{table}[tb]
\footnotesize
\caption{...}
\label{...}
\centering
\begin{tabular}{lp{0.5\textwidth}}
...
\end{tabular}
\end{table}
```

produce la tabella 43. (Se ne confronti il risultato con la tabella 34 a pagina 100.)

Ruotare una tabella

Per ruotare di 90° una tabella è utile il pacchetto `rotating`, che definisce l'ambiente `sidewaystable` da usare nel modo seguente:

```
\begin{sidewaystable}
\caption{...}
\label{...}
\centering
\begin{tabular}
...
\end{tabular}
\end{sidewaystable}
```

Si noti che una tabella di questo tipo è mobile e occupa *sempre* una pagina a sé, come mostra la tabella 9 a pagina 34. Per ruotare le immagini il pacchetto definisce l'analogo ambiente `sidewaysfigure`.

Tablelle su più pagine

Una tabella *deve* sempre poter stare in una sola pagina: se è più lunga, \LaTeX taglia le parti in eccesso. Il pacchetto `longtable` può risolvere il problema e ripartire una tabella su più pagine, anche se oggetti di questo tipo, si ricordi, andrebbero usati *soltanto se inevitabili*.

A differenza di altri pacchetti simili (come `supertabular` e `xtab`) di solito `longtable` riesce a usare su ogni pagina la stessa larghezza di riga, corrispondente a quella della riga più lunga della tabella: il risultato finale potrebbe richiedere alcune composizioni successive. Si noti che anche se può avere una didascalia, una tabella `longtable` *non* è mobile: se ne valutino sempre con attenzione opportunità e posizione, dunque, e se dovesse andare a pagina nuova molto prima della fine di quella corrente si sia pronti a riformulare il testo nei dintorni per bilanciare il risultato.

La tabella 44, che mostra il pacchetto all'opera, riporta *tutte* le parti di una `longtable` nell'ordine in cui vanno scritte, ma si ricordi che non sono obbligatorie né si deve abusarne: la didascalia non sempre è necessaria, e anche intestazioni e piedi sono facoltativi, ma si consiglia di scrivere almeno uno dei due per segnalare al lettore che la stessa tabella occupa più di una pagina.

Tabella 44: Esempio di tabella ripartita su più pagine.

Comando	Effetto
<code>\begin{longtable}</code>	Comincia la tabella. L'ambiente si comporta come <code>tabular</code> , ma in più, dopo aver composto ogni riga controlla l'altezza complessiva della tabella: se supera quella della pagina, vengono inseriti automaticamente il contenuto del piede (<i>foot</i>) e il comando <code>\end{tabular}</code> , e la tabella continua su una nuova pagina con l'intestazione scelta (<i>head</i>).
<code>[\langle carattere \rangle]</code>	Va eventualmente specificato tra l'apertura dell'ambiente e il preambolo della tabella per impostarne la posizione sulla pagina (c se al centro, l se a sinistra e r se a destra). Non indicandolo, la tabella viene centrata per impostazione predefinita.
<code>\{\langle preambolo \rangle\}</code>	È il consueto preambolo nel quale indicare i descrittori già esaminati nei paragrafi precedenti.
<code>\caption{\langle didascalia \rangle \\\</code>	Se specificato, assegna alla tabella una didascalia, anche nella versione <code>[\langle didascalia breve \rangle]</code> (si veda il paragrafo 6.2.2 a pagina 93).
<code>\label{\langle etichetta \rangle} \\\</code>	Assegna alla tabella un'etichetta, utile per i riferimenti incrociati (si veda lo stesso paragrafo).

Continua nella prossima pagina

Continua dalla pagina precedente

Comando	Effetto
<code><intestazione iniziale> \\</code> <code>\endfirsthead</code>	Specifica l' <i><intestazione iniziale></i> , cioè l'intestazione della tabella nella prima pagina in cui compare.
<code><intestazione normale> \\</code> <code>\endhead</code>	Specifica l' <i><intestazione normale></i> , cioè l'intestazione della tabella dalla seconda pagina in poi (<i>Continua dalla pagina precedente</i> , per esempio).
<code><piede normale> \\</code> <code>\endfoot</code>	Specifica il <i><piede normale></i> , cioè il testo che deve comparire alla fine di ogni pagina (<i>Continua nella prossima pagina</i>).
<code><piede finale> \\</code> <code>\endlastfoot</code>	Specifica il <i><piede finale></i> , cioè il testo che deve apparire subito dopo l'ultima riga della tabella (<i>Si conclude dalla pagina precedente</i>).
<code><corpo della tabella></code>	Qui si mette il contenuto della tabella, separando le colonne e terminando le righe come al solito.
<code>\end{longtable}</code>	Termina la tabella.

Si conclude dalla pagina precedente

Il codice seguente ha prodotto la tabella 44 nella pagina precedente:

```

\begin{longtable}{lp{0.48\textwidth}}
% intestazione iniziale
\caption{Esempio di tabella ripartita su più pagine.}
\label{tab:longtable} \\
\toprule
Comando & Effetto \\
\midrule
\endfirsthead
% intestazione normale
\multicolumn{2}{l}{\footnotesize\itshape
Continua dalla pagina precedente} \\
\toprule
Comando & Effetto \\
\midrule
\endhead
% piede normale
\midrule
\multicolumn{2}{r}{\footnotesize\itshape
Continua nella prossima pagina} \\
\endfoot
% piede finale
\bottomrule
\multicolumn{2}{r}{\footnotesize\itshape
Si conclude dalla pagina precedente} \\
\endlastfoot
% corpo della tabella
... & ... \\
...
... & ... \\
\end{longtable}

```

Per evitare errori nella composizione, si ricordi di terminare con `\\` le righe di intestazioni, piedi e didascalia (o etichetta, se presente). Si noti, infine, che in casi particolarissimi si può ruotare anche una `longtable` mettendola nell'ambiente `landscape` del pacchetto `pdflscape` (se ne veda la documentazione), ma come al solito si raccomanda di non abusare di questa possibilità.

Per gli approfondimenti si consiglia la lettura di [Gregorio, 2010].

6.4 FIGURE

Le figure rientrano tra gli argomenti più studiati dalle guide a \LaTeX , tanto che ne esistono di specifiche, cui si rimanda per gli approfondimenti.

Questi oggetti presentano almeno due problemi diversi, riguardanti:

1. il *tipo di file* da introdurre nel documento (verrà trattato in questo paragrafo);
2. la *collocazione* della figura sulla pagina (verrà trattato nel paragrafo successivo).

Di qui in avanti si dà per caricato il pacchetto `graphicx`.

6.4.1 Immagini vettoriali e bitmap

Si possono dividere le figure in due grandi classi: le immagini *vettoriali* e le immagini *bitmap* [Mori, 2007].

Immagini vettoriali

Le immagini vettoriali sono descritte da forme, possono essere scalate e deformate senza perdere in definizione e sono adatte soprattutto per grafici e schemi, argomento non considerato in questa guida per via della sua complessità (gli strumenti più diffusi per produrre grafici e schemi con \LaTeX sono l'ambiente standard `picture`, i pacchetti `PGF/TikZ`, `PSTricks`, `Xy-pic` e i programmi `METAPOST` e `Asymptote`). Molto più semplicemente, le si può aggiungere al documento dopo averle preparate a parte con programmi specifici. I formati vettoriali più noti e diffusi sono il `PDF`, il `PS` e il suo parente stretto `EPS` e l'`SVG`, usato specialmente per le applicazioni Web.

Il paragrafo 6.4.4 nella pagina seguente raccoglie alcuni programmi per la grafica vettoriale

Immagini bitmap

Le immagini bitmap sono matrici di pixel colorati, di solito perdono in definizione se ingrandite o rimpicciolite e sono più adatte a fotografie e icone. I formati bitmap sono numerosissimi e comprendono il `JPEG`, molto diffuso in ambito fotografico e nella grafica con colori morbidi, il `PNG`, adatto per grafica con colori decisi, il `GIF`, il `TIFF`. Alcuni di essi sono compressi per sfruttare al meglio la ridondanza d'informazione.

6.4.2 Convertire i formati

Prima ancora di includere le immagini nel documento bisogna produrle nel formato più adatto al proprio scopo. È inutile registrare una figura come `JPEG` per poi convertirla in `PDF`, perché la conversione include semplicemente

Tabella 45: Alcuni programmi utili per lavorare con \LaTeX (le lettere G, C, R e V indicano rispettivamente le funzioni di grafica vettoriale, conversione dei formati, ritaglio immagini e visualizzazione; il simbolo • indica che la funzione è disponibile).

Programma	G	C	R	V
Inkscape	•	•		•
Xfig e WinFIG	•			
Gnuplot	•			
Mathematica	•			
OmniGraffle	•			
GIMP		•	•	•
Adobe Acrobat			•	•
Adobe Reader				•
Anteprima		•	•	•
ImageMagick		•		
eps2pdf e epstopdf		•		
Ghostview e GSview		•		•

il file bitmap in una “cornice” PDF senza migliorarne in alcun modo la qualità. È sbagliato anche fare l’opposto, perché così si perdono le informazioni sulla geometria della figura, abbassandone la qualità. Nonostante questo, si potrebbero avere a disposizione soltanto immagini in formati *non* adatti a \LaTeX , e allora la conversione sarebbe davvero necessaria.

A questo proposito si ricordi bene che \LaTeX accetta immagini PDF, JPEG, PNG ed EPS. Il pacchetto `bmsize`, infine, permette di inserire anche immagini BMP, GIF e TIFF.

Il paragrafo 6.4.4 descrive alcuni programmi per convertire i diversi formati.

6.4.3 Ritagliare le immagini

Uno dei parametri più importanti di una figura è l’informazione sulle dimensioni del rettangolo circoscritto a essa (*bounding box*). Questo contorno determina la grandezza effettiva dell’immagine e serve a \LaTeX per calcolare lo spazio da riservarle sulla pagina. Idealmente, il contorno dovrebbe coincidere con il limite dell’immagine, ma talvolta le figure sono circondate da un invisibile bordo bianco più o meno ampio, fonte di non pochi problemi estetici: la figura appare sulla pagina troppo piccola o non centrata o circondata da eccessivi margini verticali, per esempio, anche se \LaTeX la sta trattando nel modo corretto.

La primissima cosa da verificare, quindi, è che le dimensioni della *bounding box* siano corrette, aprendo la figura con un programma opportuno (come Adobe Reader o GIMP) e attivando la visualizzazione del contorno che, se scorretto, va ridimensionato. Se il problema riguarda poche figure si può risolvere a mano, ma se i file da ottimizzare sono molti, vanno corretti all’origine (magari configurando ad hoc il programma usato per produrli).

6.4.4 Alcuni programmi utili

La composizione asincrona di \LaTeX ha anche un vantaggio da non trascurare: permette di usare sempre il prodotto migliore. Per ciascuna operazione sul documento, infatti, l’utente può usare un programma specializzato, ciò che non sarebbe permesso con un software “tuttofare” come un editor di testi tradizionale. La tabella 45 raccoglie, senza pretese di completezza,

alcuni programmi utili per lavorare con \LaTeX (ma preziosi anche in molte altre occasioni), specificandone le funzioni principali. Una veloce ricerca in Rete permette di recuperarli.

6.4.5 Includere le immagini nel documento

Il pacchetto `graphicx`, che in genere non richiede opzioni, gestisce il trattamento delle immagini con \LaTeX . Il comando `\includegraphics`, la cui sintassi completa è

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

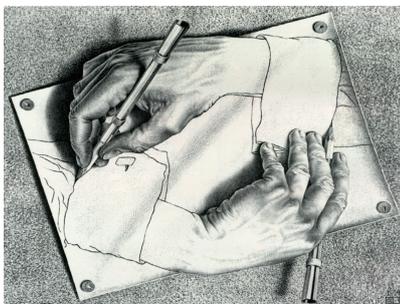
le include nel documento. Si osservi che:

- nell'argomento facoltativo ci vanno le opzioni che regolano l'aspetto della figura sulla pagina nella forma `<chiave>=<valore>` (si veda la tabella 46 nella pagina successiva);
- nell'argomento obbligatorio ci va il nome dell'immagine *senza* specificarne l'estensione.

Gli esempi seguenti mostrano il pacchetto all'opera.

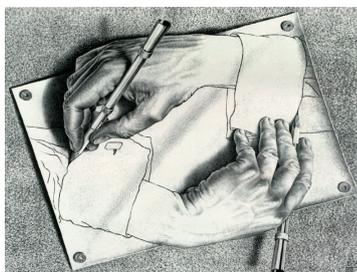
L'utente può assegnare all'immagine una larghezza (`width`)

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



o un'altezza (`height`) a piacere:

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



Ancora, la può ridimensionare nel suo complesso

```
\includegraphics[scale=0.10%  
{mani}
```



o ruotare di un certo angolo in entrambi i sensi:

```
\includegraphics[width=0.5%  
\textwidth,angle=45]{mani}
```

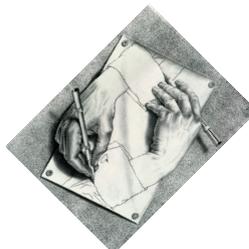


Tabella 46: Principali chiavi di `graphicx`.

Chiave	Agisce su
<code>width</code>	Larghezza
<code>height</code>	Altezza
<code>scale</code>	Larghezza e altezza
<code>angle</code>	Orientamento

Si noti che:

- per i motivi già spiegati nel paragrafo 6.1 a pagina 91, le figure devono avere dimensioni *relative*, cioè essere larghe una frazione di `\textwidth` e alte una frazione di `\textheight` (l'altezza della gabbia del testo);
- il valore di scalatura si esprime con un numero decimale;
- l'angolo di rotazione si esprime con un numero (negativo, se la rotazione è oraria) nell'intervallo 0-360;
- se s'intende assegnare all'immagine sia una larghezza sia un'altezza determinate, si ricordi di specificare anche la chiave `keepaspectratio` per evitare di distorcerla.

6.5 DISPOSIZIONI PARTICOLARI

Questo paragrafo presenta alcune possibilità per organizzare la disposizione degli oggetti mobili sulla pagina in modi diversi da quelli predefiniti. Quanto si dirà vale sia per le tabelle sia per le figure.

6.5.1 Didascalie laterali: `sidecap`

Il pacchetto `sidecap` produce la didascalia *accanto* all'oggetto (di solito una figura, più raramente una tabella) anziché sopra o sotto. Le opzioni fondamentali del pacchetto sono le seguenti:

outercaption La didascalia è posta nel margine esterno della pagina (a sinistra in quelle pari e a destra in quelle dispari). È l'opzione predefinita.

innercaption La didascalia è posta nel margine interno della pagina (a destra in quelle pari e a sinistra in quelle dispari).

leftcaption, rightcaption La didascalia è sempre posta a sinistra o a destra dell'oggetto, rispettivamente.

ragged, raggedright, raggedleft Gestiscono l'allineamento delle didascalie brevi.

Il pacchetto definisce due nuovi ambienti `SCfigure` e `SCTable` (analoghi agli ambienti `figure` e `table`) che prevedono due argomenti facoltativi,

```
\begin{SCfigure}[larghezza relativa][collocazione]
```

e

```
\begin{SCTable}[larghezza relativa][collocazione]
```

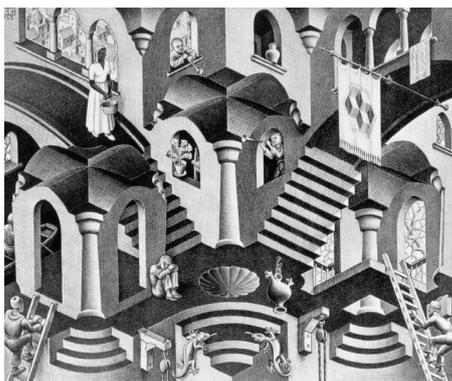


Figura 8: Esempio d'uso di sidecap (l'immagine riproduce la litografia *Concavo e convesso* di M. Escher).

dove:

- $\langle larghezza\ relativa \rangle$ indica il rapporto fra larghezza della didascalia e larghezza della figura (o della tabella) corrispondente. Un valore elevato di $\langle larghezza\ relativa \rangle$ (50, per esempio) assegna alla didascalia la massima larghezza possibile. Il valore predefinito è 1.
- $\langle collocazione \rangle$ indica le preferenze di collocazione degli ambienti mobili, da usare come al solito. Il valore predefinito è `tbp`.

Esistono anche gli ambienti `SCfigure*` e `SCTable*` (la cui sintassi è la stessa di `SCfigure` e `SCTable`) che in un documento a più colonne permettono di ottenere rispettivamente una figura o una tabella mobile (con didascalia laterale) estesa su tutta la pagina.

La figura 8 è stata inclusa nel documento con un codice del tipo:

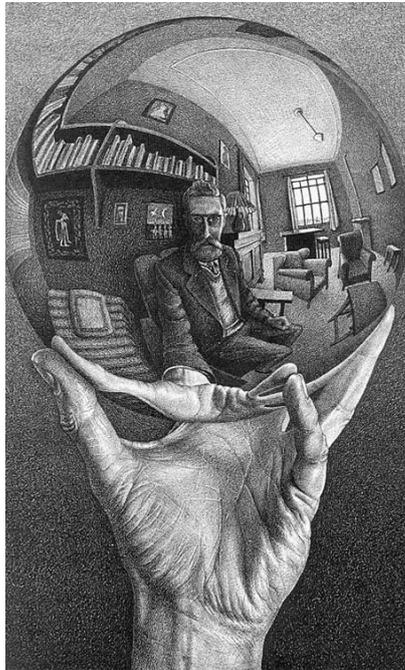
```
\begin{SCfigure}[]
\centering
\includegraphics[width=0.5\textwidth]{ConcavoConvesso}
\caption{Un esempio di figura con didascalia laterale.}
\label{fig:sidecap}
\end{SCfigure}
```

6.5.2 Oggetti multipli: subfig

Il pacchetto `subfig` (che richiede `caption`) gestisce con `\subfloat` più (sotto)figure (come la figura 9 nella pagina seguente) o (sotto)tabelle in un unico ambiente mobile e ne regola le didascalie molto finemente.

La figura 9 nella pagina successiva è stata ottenuta con un codice del tipo:

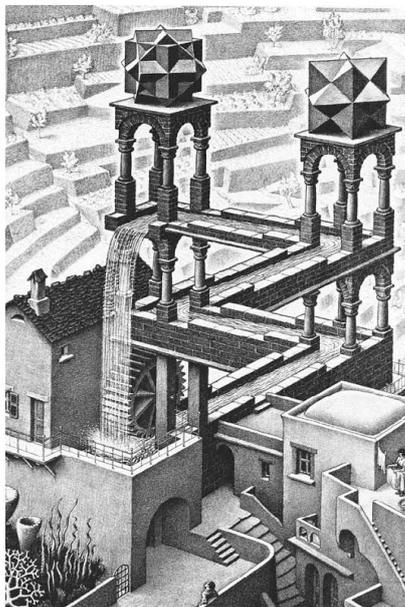
```
\begin{figure}
\centering
\subfloat[][\emph{Mano con sfera riflettente}.]{
  \includegraphics[width=.45\textwidth]{Sfera} \quad
}
\subfloat[][\emph{Belvedere}.]{
  \includegraphics[width=.45\textwidth]{Belvedere} \quad \quad
}
\subfloat[][\emph{Cascata}.]{
  \includegraphics[width=.45\textwidth]{Cascata} \quad
}
\subfloat[][\emph{Salita e discesa}.]{
  \includegraphics[width=.45\textwidth]{SalitaDiscesa}
}
\caption{Alcune litografie di M.-Escher.}
\label{fig:subfig}
\end{figure}
```



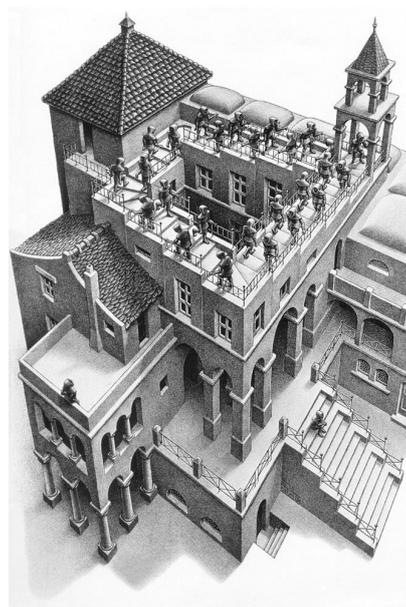
(a) Mano con sfera riflettente.



(b) Belvedere.



(c) Cascata.



(d) Salita e discesa.

Figura 9: Esempio d'uso di subfig (le immagini riproducono alcune litografie di M. Escher).

Si noti che:

- nel primo argomento facoltativo di `\subfloat`, se usato, si mette la didascalia breve da mandare nel relativo indice (`\listoffigures` o `\listoftables`);
- nel secondo ci va la didascalia che comparirà effettivamente sulla pagina;
- per riferirsi a un sottooggetto in particolare da altre parti del documento, `\label` va dato *dentro* il secondo argomento facoltativo immediatamente dopo la sottodidascalia.

6.5.3 Oggetti immersi nel testo: wrapfig

In alcune circostanze può essere desiderabile “avvolgere” un oggetto con del testo, magari anche solo per movimentare la pagina. Risolve il problema il pacchetto `wrapfig`, particolarmente indicato perché interagisce correttamente con `caption` per personalizzare la didascalia.

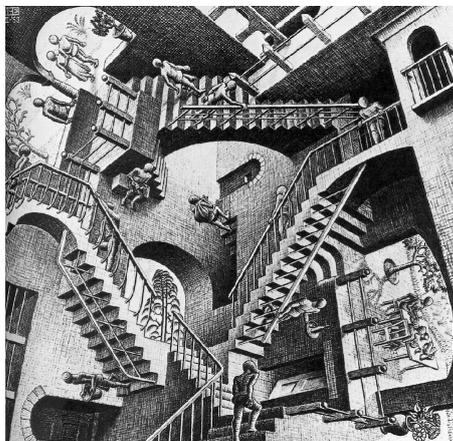


Figura 10: Esempio d’uso di `wrapfig` (l’immagine riproduce la litografia *Relatività* di M. C. Escher).

Il pacchetto definisce l’apposito ambiente `wrapfloat`, nel quale mettere l’oggetto con i comandi consueti. Ragioni estetiche impongono di circondarlo soltanto con testo continuo (come qui), rimandando più oltre eventuali altri oggetti o ambienti particolari. Tuttavia, anche operando correttamente il pacchetto non garantisce un risultato ottimale già alla prima composizione: potrebbero essere necessari numerosi aggiustamenti manuali della pagina, più o meno consistenti.

Come si può notare, una “wrapfig” correttamente ottenuta dà un risultato altamente professionale, ma si raccomanda di sfruttare questa possibilità soltanto in circostanze *davvero* eccezionali, cioè praticamente *mai*. La buona riuscita dell’operazione richiede condizioni particolari: composizione a piena pagina e gabbia sufficientemente ampia da poter accogliere una figura larga la sua metà: in caso contrario, la colonna di testo accanto all’immagine risulterà troppo stretta e inevitabilmente solcata da ruscelli. La regola da seguire rimane la stessa: includere tutti gli oggetti fuori testo e lasciar fare a \LaTeX . Soltanto a lavoro ultimato, quando il risultato finale è davvero soddisfacente, sarà possibile ricollocarne qualcuno secondo il proprio gusto.

La figura 10 è stata ottenuta con un codice del tipo:

```
\dots “Avvolgere” un oggetto con del testo è molto semplice.
```

```
\begin{wrapfloat}{figure}{I}{0pt}
\includegraphics[width=0.5\textwidth]{Relativo}
\caption{Esempio di figura “avvolta” da un testo.}
\end{wrapfloat}
```

Può rendersi necessario un po’ di lavoro per ottenere un’impaginazione ottimale. \dots

Tabella 47: Preferenze di collocazione dell'ambiente `wrapfloat`.

r, R	Sul lato destro del testo (<i>right</i>)
l, L	Sul lato sinistro del testo (<i>left</i>)
i, I	Sul margine interno (<i>inner</i>)
o, O	Sul margine esterno (<i>outer</i>)

Come si vede nell'esempio precedente, l'ambiente `wrapfloat` richiede tre argomenti obbligatori:

```
\begin{wrapfloat}{\langle oggetto \rangle}{\langle collocazione \rangle}{\langle larghezza \rangle}
```

Dove:

- `\langle oggetto \rangle` indica il tipo di *oggetto* da includere (`figure` o `table`, da non confondere con gli ambienti omonimi);
- `\langle collocazione \rangle`, che dice a \LaTeX dove mettere l'oggetto sulla pagina, accetta *una sola* delle otto opzioni della tabella 47, in maiuscolo o in minuscolo a seconda che si voglia mettere l'oggetto "esattamente qui nel testo" o si voglia creare un oggetto mobile, rispettivamente;
- `\langle larghezza \rangle` specifica la larghezza dell'oggetto che, se nulla (`0pt`), equivale all'opzione assegnata a `\includegraphics`.

7 | BIBLIOGRAFIA

La bibliografia è da sempre uno degli aspetti più delicati di un documento, e \LaTeX aiuta anche in questo caso, definendo tutti gli strumenti per realizzarla e gestirla con efficienza e flessibilità. L'argomento è tecnico, perciò è bene chiarire da subito alcuni termini usati (anche in forma abbreviata) in questo capitolo:

BIBLIOGRAFIA Elenco di opere scritte o di altro tipo che di solito occupa una sezione autonoma del documento con un titolo (in genere) omonimo.

RIFERIMENTO BIBLIOGRAFICO Serie di dati che permette di identificare ed eventualmente reperire un'opera. L'insieme dei riferimenti bibliografici costituisce la bibliografia di un documento.

STILE BIBLIOGRAFICO Modalità generale adottata per presentare al lettore i riferimenti bibliografici.

CITAZIONE BIBLIOGRAFICA Indicazione sintetica, data nel corpo del documento, che rinvia il lettore a un riferimento bibliografico.

SCHEMA DI CITAZIONE Modalità generale adottata per presentare al lettore una citazione bibliografica.

Con \LaTeX si può creare la bibliografia in due modi, essenzialmente:

- a mano, con l'ambiente `thebibliography`;
- automaticamente, con il pacchetto `biblatex`.

7.1 BIBLIOGRAFIA MANUALE

L'ambiente `thebibliography` gestisce la bibliografia di un documento molto facilmente, ma non è altrettanto flessibile. La sintassi generale è:

```
\begin{thebibliography}{\langle etichetta più lunga \rangle}
\bibitem[\langle etichetta personalizzata \rangle]{\langle chiave di citazione \rangle}
...
\end{thebibliography}
```

dove:

- $\langle etichetta \text{ più lunga} \rangle$ può essere un numero (di solito 9 se la bibliografia comprende meno di dieci opere, 99 se almeno dieci ma meno di cento, eccetera) oppure del testo (nel caso di etichette personalizzate: si scrive allora l'etichetta più lunga);
- `\bibitem` va premesso a ogni riferimento bibliografico;
- $\langle etichetta \text{ personalizzata} \rangle$ è un'etichetta personalizzata che *eventualmente* comparirà nella bibliografia e nelle citazioni al posto del numero predefinito;
- $\langle chiave \text{ di citazione} \rangle$ è l'etichetta *univoca* per citare la fonte nel documento (si consiglia di usare la sintassi $\langle autore \rangle: \langle titolo \rangle$ analoga a quella di `\label`).

Lo si vede all'opera nell'esempio seguente:

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

\bibitem{mori:tesi}
Mori, Lapo Filippo (2007),
"Scrivere la tesi di laurea
con \LaTeXe", \Ars-(3).
\end{thebibliography}
```

BIBLIOGRAFIA

- [1] Eco, Umberto (1977), *Come si fa una tesi di laurea*, Bompiani, Milano.
- [2] Mori, Lapo Filippo (2007), "Scrivere la tesi di laurea con $\LaTeX 2\epsilon$ ", *ArsTeXnica* (3).

Si noti che:

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

Per citare un riferimento bibliografico nel testo si usa il comando `\cite`:

```
\cite[informazioni][chiave di citazione]
```

dove:

- *informazioni* sono ulteriori indicazioni (numero di pagina o di capitolo) che eventualmente si possono dare per completare la citazione;
- *chiave di citazione* si spiega da sé.

I prossimi esempi lo mostrano all'opera:

Si veda `\cite{eco:tesi}` per maggiori dettagli.

Si veda [1] per maggiori dettagli.

Si veda `\cite[p.~7]{eco:tesi}` per maggiori dettagli.

Si veda [1, p. 7] per maggiori dettagli.

Per mandare nell'indice generale il titolo della bibliografia del documento, *immediatamente prima* di aprire thebibliography vanno date sequenze di comandi diverse in base alla classe in uso:

```
\cleardoublepage
%\phantomsection
\addcontentsline{toc}{chapter}{\bibname}
```

se la classe è book o report, e

```
%\clearpage
%\phantomsection
\addcontentsline{toc}{section}{\refname}
```

se la classe è article, dove:

- `\cleardoublepage` fa cominciare la bibliografia in una pagina nuova dispari e assegna alla corrispondente voce nell'indice il numero di pagina corretto;
- `\clearpage` va dato per assicurare la corretta assegnazione del numero di pagina alla voce nell'indice *solo se* a fine composizione il corpo del documento terminasse esattamente a fine pagina e la bibliografia cominciasse all'inizio di una pagina nuova (in tal caso si decommenti la riga corrispondente);
- `\phantomsection` va dato *solo se* è caricato anche `hyperref` (in tal caso si decommenti la riga corrispondente);
- `chapter` e `section` indicano il livello della sezione bibliografica (un capitolo e un paragrafo, rispettivamente);
- `\bibname` e `\refname` producono nell'indice generale del documento le voci *Bibliografia* e *Riferimenti bibliografici* rispettivamente.

7.2 BIBLIOGRAFIA AUTOMATICA

L'ambiente `thebibliography` produce risultati dignitosi, ma a prezzo di alcuni inconvenienti non da poco:

- l'ordinamento alfabetico dei riferimenti nella bibliografia non è automatico;
- bisogna scrivere una bibliografia per ogni documento, anche se c'è solo qualche opera di differenza dall'uno all'altro;
- se si aggiorna un documento, bisogna modificare le bibliografie di *tutti* i documenti in cui esso è citato;
- per cambiare lo stile bibliografico bisogna modificare a mano *tutte* le voci, una per una.

Come si noterà, anziché compilare bibliografie indipendenti è decisamente meglio generarle *automaticamente* creando *una volta per tutte* un solo database: sistemandolo in un'opportuna cartella dell'*albero personale* (per esempio, `$HOME/texmf/bibtex/bib/mieibib`) basterà aggiornarlo all'occorrenza.

In questo paragrafo si presentano gli elementi fondamentali del pacchetto `biblatex`, un potentissimo strumento per gestire automaticamente la bibliografia e personalizzare ogni aspetto degli stili bibliografici e di citazione con poche operazioni.

Per funzionare correttamente, il pacchetto richiede di caricare *anche* `babel` (o `polyglossia`, se si compone con `XYLATEX`, si veda il paragrafo [B.2.2](#) a pagina 204) e `csquotes` con le opzioni indicate di seguito, rispettando l'ordine dei comandi:

```
\usepackage[autostyle,italian=guillemets,<altre opzioni>]{csquotes}
\usepackage[<opzioni>,backend=biber]{biblatex}
```

Si noti che:

- `autostyle` adatta lo stile delle citazioni alla lingua corrente del documento;

- `italian=guillemets` racchiude automaticamente tra virgolette caporali i campi che prevedono le virgolette;
- `backend=biber` dice a biblatex che s'intende usare Biber come motore bibliografico (si veda il paragrafo 7.2.1).

Si ricordi infine biblatex *non* è compatibile con i pacchetto ucs e natbib.

7.2.1 Biber

Dalla versione 2012 di T_EX Live, il nuovo motore bibliografico predefinito da biblatex è Biber. B_IB_TE_X è ancora utilizzabile, ma il fatto che lavori con una versione obsoleta di biblatex suggerisce di preferire senz'altro il primo, il cui sviluppo procede parallelamente a quello del pacchetto presentato in questo capitolo. Ed è questa la scelta qui raccomandata.

Gli editor per L^AT_EX presentati nel paragrafo 2.1.1 a pagina 11, tuttavia, sono ancora preimpostati per lavorare con B_IB_TE_X. In attesa che la situazione si stabilizzi, si danno di seguito le semplici istruzioni per passare a Biber:

TEXSHOP Si segua il percorso TeXShop → Preferenze... → Motore e nella riga BibTeX Engine si sostituisca biber a bibtex.

T_EXWORKS Si seguano le istruzioni contenute in [Gregorio, 2011].

TEXMAKER Si segua il percorso texmaker → Preferences... e nella riga Bib(la)-tex si sostituisca biber a bibtex.

TEXSTUDIO Si segua il percorso opzioni → Configure TeXstudio... e nella riga BibTeX si sostituisca biber a bibtex.

EMACS Si veda la documentazione di Biber (consultabile con `texdoc biber`).

7.2.2 Database bibliografici

Un *database bibliografico* è un file da registrare con estensione `.bib` (si scrive con l'editor in uso e *non* va composto) che contiene un certo numero di *record* come i seguenti:

```
@book{eco:tesi,
  author      = {Eco, Umberto},
  title       = {Come si fa una tesi di laurea},
  publisher    = {Bompiani},
  date        = {1977},
  location    = {Milano},
}

@article{mori:tesi,
  author      = {Mori, Lapo Filippo},
  title       = {Scrivere la tesi di laurea con \LaTeXe},
  journaltitle = {\Ars},
  number      = {3},
  date        = {2007},
}

>manual{mori:poesie,
  author      = {Mori, Lapo Filippo},
  title       = {Scrivere poesie con \LaTeX},
  date        = {2007},
  url         = {http://www.guitex.org/},
}
```

```
@online{wiki:latex,
  title      = {\LaTeX} su Wikipedia},
  date       = {2012},
  url        = {http://it.wikipedia.org/wiki/LaTeX},
  sortkey    = {wiki},
  label      = {wiki},
}
```

Si osservi che:

- Ogni record corrisponde a un riferimento bibliografico, il cui tipo va indicato come prima cosa (indifferentemente in maiuscolo o minuscolo) subito dopo @. Si tratta in questo caso di un libro (record di tipo @book), di un articolo (@article), di un manuale (@manual) e di un sito Web (@online).
- S'indica poi una *chiave*, di solito nella forma $\langle \text{cognome} \rangle : \langle \text{parola chiave} \rangle$ (dove $\langle \text{parola chiave} \rangle$ è un segnaposto univoco della fonte da citare), da usare nell'argomento dei comandi per le citazioni.
- Infine, si riempiono i *campi* che definiscono l'opera (autore, titolo, eccetera). Di qualunque tipo sia, un campo ha la forma

$\langle \text{nome del campo} \rangle = \{ \langle \text{contenuto del campo} \rangle \},$

e, come si vede, va terminato con la virgola, *anche se è l'ultimo*, pena un errore.

Per indicare a L^AT_EX quale o quali database usare per comporre la bibliografia si dà nel preambolo il comando

`\addbibresource{ $\langle \text{nome del database} \rangle$.bib}`

ripetendolo per ogni base di dati e *specificando l'estensione* .bib.

7.2.3 Record e campi

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

- *obbligatorio*, cioè indispensabile a biblatex per comporre il riferimento bibliografico;
- *opzionale*, cioè non indispensabile ma utile, se specificato, per aggiungerci informazioni.

I record e i campi disponibili con biblatex sono numerosissimi e per tutte le esigenze; il loro elenco completo si trova nella documentazione del pacchetto.

Record

Di seguito si elencano i principali *record* standard riconosciuti da biblatex, indicando di ognuno campi obbligatori e principali campi opzionali (il loro significato verrà spiegato nel paragrafo successivo). In particolari situazioni, si possono sostituire i campi in corsivo con degli altri (si veda il paragrafo 7.2.7 a pagina 133).

@article Articolo apparso in una rivista o in un giornale.

Campi obbligatori: author, title, journaltitle, date.

Campi opzionali: editor, series, volume, number, month, pages, note, url.

- @book** Libro regolarmente pubblicato da una casa editrice.
Campi obbligatori: *author*, *title*, *date*.
Campi opzionali: *editor*, *volume*, *series*, *note*, *publisher*, *location*, *number*, *url*.
- @collection** Raccolta di contributi indipendenti di autori diversi.
Campi obbligatori: *editor*, *title*, *date*.
Campi opzionali: *volume*, *series*, *number*, *publisher*, *location*, *note*, *chapter*, *pages*, *url*.
- @proceedings** Atti di un convegno o conferenza.
Campi obbligatori: *editor*, *title*, *date*.
Campi opzionali: *volume*, *series*, *number*, *note*, *location*, *publisher*, *organization*, *chapter*, *pages*, *url*.
- @inbook** Parte di un libro con un titolo proprio.
Campi obbligatori: *author*, *title*, *booktitle*, *date*.
Campi opzionali: *editor*, *volume*, *series*, *note*, *publisher*, *location*, *chapter*, *number*, *pages*, *url*.
- @bookinbook** Opera pubblicata originariamente come libro autonomo e ristampata, per esempio, nell'*opera omnia* di un autore. Per i campi, si veda la voce precedente.
- @incollection** Parte di una raccolta con un titolo proprio.
Campi obbligatori: *author*, *title*, *booktitle*, *date*.
Campi opzionali: *volume*, *series*, *number*, *publisher*, *location*, *note*, *chapter*, *pages*, *url*.
- @inproceedings** Articolo negli atti di un convegno o intervento in una conferenza.
Campi obbligatori: *author*, *editor*, *title*, *booktitle*, *date*.
Campi opzionali: *volume*, *series*, *number*, *note*, *location*, *publisher*, *organization*, *chapter*, *pages*, *url*.
- @booklet** Libro distribuito in proprio.
Campi obbligatori: *author* o *editor*, *title*, *date*.
Campi opzionali: *howpublished*, *type*, *note*, *location*, *chapter*, *pages*, *url*.
- @manual** Documentazione tecnica.
Campi obbligatori: *author* o *editor*, *title*, *date*.
Campi opzionali: *type*, *version*, *series*, *number*, *organization*, *note*, *publisher*, *location*, *chapter*, *pages*, *url*.
- @report** Relazione pubblicata da università, scuola o altra istituzione.
Campi obbligatori: *author*, *title*, *type*, *institution*, *date*.
Campi opzionali: *number*, *note*, *location*, *chapter*, *pages*, *url*.
- @thesis** Tesi di laurea o di dottorato. Il campo *type* specifica il tipo di tesi.
Campi obbligatori: *author*, *title*, *type*, *institution*, *date*.
Campi opzionali: *note*, *location*, *chapter*, *pages*, *url*.
- @online** Risorsa disponibile su Internet.
Campi obbligatori: *author* o *editor*, *title*, *date*, *url*.
Campi opzionali: *note*, *organization*, *date*.
- @unpublished** Documento con autore e titolo, ma non pubblicato.
Campi obbligatori: *author*, *title*, *date*.
Campi opzionali: *howpublished*, *note*, *date*, *url*.

@misc Record da usare quando nessuno degli altri è appropriato.

Campi obbligatori: *author* o *editor*, *title*, *date*.

Campi opzionali: *howpublished*, *type*, *organization*, *location*, *note*, *date*, *url*.

Campi

Di seguito si riporta l'elenco dei principali *campi* riconosciuti da biblatex.

author Cognome e nome dell'autore (o degli autori, nel formato descritto più avanti) separati con la virgola.

booktitle Titolo dell'opera che contiene la fonte (se essa è solo una parte della pubblicazione).

chapter Numero del capitolo (o di una qualunque parte del documento).

date Anno di pubblicazione della fonte.

edition Numero di edizione della fonte.

editor Cognome e nome del curatore (o dei curatori) separati con la virgola.

howpublished Genere della pubblicazione.

institution Nome dell'università o dell'istituzione.

journaltitle Nome della rivista o del giornale.

label Etichetta per le citazioni, nel caso in cui manchino i dati necessari per formare l'etichetta "regolare" (si veda il paragrafo 7.2.7 a pagina 133).

location Indirizzo di editore (*publisher*) o istituzione (*institution*).

month Mese di pubblicazione dell'opera. Deve essere un numero intero; per esempio, non si scrive *month*={January}, ma *month*={1}.

note Informazioni supplementari che possono aiutare il lettore a identificare meglio l'opera.

number Numero della rivista, del giornale o della raccolta.

organization Organizzazione che pubblica il documento o patrocina la conferenza.

pages Uno o più numeri di pagina.

publisher Nome dell'editore.

series Nome della collezione di cui la fonte fa parte.

sortkey Ordina alfabeticamente le voci bibliografiche prive dell'indicazione di autore o curatore (si veda il paragrafo 7.2.7 a pagina 133).

title Titolo della fonte.

type Tipo di manuale, relazione o tesi.

url Indirizzo Internet di riferimento per la fonte.

volume Numero di volume della fonte.

Alcune precisazioni

Gli stili *predefiniti* di biblatex producono il titolo dell'opera rispettando le maiuscole e le minuscole. Un titolo come

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

verrà prodotto esattamente com'è scritto.

Un comando che produce testo presente in un campo non va messo tra graffe se si lavora con uno stile predefinito:

```
title = {L'arte di scrivere con \LaTeX}
```

ma potrebbe essere necessario farlo se lo stile è personalizzato:

```
title = {L'arte di scrivere con {\LaTeX}}
```

Più nomi in un campo `author` o `editor` vanno separati con `and`:

```
author = {Mori, Lapo Filippo and Himmelmann, Maurizio}
```

Se l'elenco dei nomi degli autori o dei curatori è troppo lungo, può essere concluso da `and others`, che di regola viene reso da biblatex con *et al.*:

```
author = {Gregorio, Enrico and Mori, Lapo Filippo and  
Pantieri, Lorenzo and others}
```

I cognomi multipli si scrivono separandoli dal nome in questo modo:

```
author = {Levi Montalcini, Rita}
```

I cognomi preceduti da particelle con iniziale *minuscola* (*von* o *van*, per esempio) richiedono qualche attenzione. S'immagini che l'autore sia *Ludwig van Beethoven*: per impostazione predefinita gli stili standard ignorano la particella precedente il cognome, ordinandolo alfabeticamente di conseguenza. L'opzione `useprefix` evita questo comportamento e dice a biblatex di considerare il cognome per intero secondo l'uso italiano. (Si noti che l'opzione `useprefix=false`, predefinita negli stili standard, ignora eventuali particelle *qualunque stile si usi*.)

Se la particella ha l'iniziale maiuscola, invece, l'inconveniente appena descritto non si presenta: *De Gasperi* cadrà sotto la lettera *d*.

Si noti che alcuni stili *personalizzati* di biblatex producono automaticamente il titolo in tutte lettere minuscole, con risultati a volte indesiderati. Si può risolvere il problema racchiudendo ulteriormente tra graffe solo le parole problematiche, così:

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

oppure l'intero titolo, così:

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

7.2.4 Stili bibliografici e schemi di citazione

L'aspetto dei riferimenti bibliografici e delle citazioni, che biblatex adatta automaticamente alla lingua principale del documento, si specificano in modi diversi. Il codice

```
\usepackage[bibstyle=authortitle,citestyle=verbose-trad1]{biblatex}
```

imposta per la bibliografia lo *stile bibliografico* (bibstyle) autore-titolo (authortitle), e per le citazioni uno *stile di citazione* (citestyle) che riporta per intero il riferimento solo nella prima citazione del documento e usa abbreviazioni in quelle successive (verbose-trad1). Il codice

```
\usepackage[style=alphabetic]{biblatex}
```

invece, imposta con lo stile (style) alfabetico (alphabetic) entrambi i parametri. Lo schema (e quindi lo stile) più adatto da usare dipende *anche* dal tipo di lavoro che si sta scrivendo.

Stili bibliografici

Il pacchetto biblatex definisce quattro *stili bibliografici* predefiniti, i quali agiscono *nella sezione bibliografica del documento*:

- ordinano alfabeticamente le opere in base al cognome di autore o curatore;
- possono contrassegnare o meno l'opera con un'etichetta;
- sistemano diversamente i dati nei riferimenti bibliografici.

Di seguito li si descrive brevemente.

numeric Anno di pubblicazione: in fondo al riferimento.

Etichetta: numero progressivo ([1], [2], eccetera).

Uso: documenti scientifici, dove importa conoscere subito *non chi* viene citato, ma che *qualcuno* viene citato e solo poi *che cosa* questo qualcuno ha scritto.

alphabetic Anno di pubblicazione: in fondo al riferimento.

Etichetta: prima parte del cognome dell'autore e ultime due cifre dell'anno di pubblicazione ([Moro7]).

authoryear Anno di pubblicazione: presente solo nell'etichetta.

Etichetta: cognome di autore o curatore e anno di pubblicazione ([Bec-cari, 2012]).

Uso: documenti in cui è importante indicare questo due dati direttamente nella citazione.

authortitle Anno di pubblicazione: in fondo al riferimento.

Etichetta: no.

Uso: la citazione è una nota al piede contenente cognome dell'autore e titolo dell'opera. Si usa quasi esclusivamente nei documenti umanistici.

Schemi di citazione

A ciascuno stile si possono associare (non contemporaneamente) uno o più *schemi di citazione*, che ne riproducono l'effetto nelle citazioni fatte *nel corpo del documento*, ma ogni schema può essere associato a *un solo* stile bibliografico alla volta. Di seguito si descrivono quattro schemi predefiniti da biblatex e alcune loro varianti.

numeric, numeric-comp Riferimento: numerico ([1], [2], eccetera).

Da associare a: stile numeric.

Varianti: numeric-comp, che ordina e comprime le citazioni multiple.

Per esempio, si ottiene [2-4, 8] al posto di [4, 2, 8, 3].

Bibliografia

- [1] Umberto Eco. *Come si fa una tesi di laurea*. Milano: Bompiani, 1977.
- [2] Lapo Filippo Mori. "Scrivere la tesi di laurea con $\text{\LaTeX}2_{\epsilon}$ ". In: *AsTeXnica* 3 (2007).
- [3] Lapo Filippo Mori. *Scrivere poesie con \LaTeX* . 2007. URL: <http://www.guit.sssup.it/>.
- [4] *\text{\LaTeX}* su *Wikipedia*. 2010. URL: <http://it.wikipedia.org/wiki/LaTeX>.

(a) Stile numeric.

Bibliografia

[Eco77] Umberto Eco. *Come si fa una tesi di laurea*. Milano: Bompiani, 1977.

[Mor07a] Lapo Filippo Mori. "Scrivere la tesi di laurea con $\text{\LaTeX}2_{\epsilon}$ ". In: *AsTeXnica* 3 (2007).

[Mor07b] Lapo Filippo Mori. *Scrivere poesie con \LaTeX* . 2007. URL: <http://www.guit.sssup.it/>.

[Wiki10] *\text{\LaTeX}* su *Wikipedia*. 2010. URL: <http://it.wikipedia.org/wiki/LaTeX>.

(b) Stile alphabetic.

Bibliografia

Eco, Umberto (1977). *Come si fa una tesi di laurea*. Milano: Bompiani.

Mori, Lapo Filippo (2007a). "Scrivere la tesi di laurea con $\text{\LaTeX}2_{\epsilon}$ ". In: *AsTeXnica* 3.

– (2007b). *Scrivere poesie con \LaTeX* . URL: <http://www.guit.sssup.it/>.

Wiki (2010). *\text{\LaTeX}* su *Wikipedia*. URL: <http://it.wikipedia.org/wiki/LaTeX>.

(c) Stile authoryear.

Bibliografia

Eco, Umberto. *Come si fa una tesi di laurea*. Milano: Bompiani, 1977.

Mori, Lapo Filippo. "Scrivere la tesi di laurea con $\text{\LaTeX}2_{\epsilon}$ ". In: *AsTeXnica* 3 (2007).

– *Scrivere poesie con \LaTeX* . 2007. URL: <http://www.guit.sssup.it/>.

\text{\LaTeX} su *Wikipedia*. 2010. URL: <http://it.wikipedia.org/wiki/LaTeX>.

(d) Stile authortitle.

Figura 11: Esempi di stili bibliografici.

alphabetic Riferimento: misto ([Moro7]).

Da associare a: stile `alphabetic`.

authoryear, authoryear-comp Riferimento: autore, anno ([Mori, 2007]).

Da associare a: stile `authoryear`.

Varianti: `authoryear-comp`, che ordina e comprime le citazioni multiple con lo stesso autore (ed eventualmente lo stesso anno di pubblicazione). Per esempio, si ottiene [Eco 1977; Mori 2007a,b] anziché [Mori 2007b; Eco 1977; Mori 2007a].

authortitle, verbose, verbose-trad1 Riferimento: autore, titolo (Mori, *Titolo*).

Da associare a: stile `authortitle`.

Varianti: `verbose`, che cita per intero il riferimento solo la prima volta e usa una forma abbreviata quelle successive; `verbose-trad1`, variante del precedente, che a seconda del contesto, usa le formule latine *idem*, *ibidem*, *op. cit.* e *loc. cit.*

Per l'elenco completo degli schemi di citazione di `biblatex` si rimanda alla documentazione del pacchetto.

Per ogni schema di citazione c'è uno stile bibliografico adatto, che `biblatex` imposta automaticamente in base allo schema scelto. Il codice

```
\usepackage[style=authoryear-comp]{biblatex}
```

infatti, equivale a

```
\usepackage[bibstyle=authoryear,citestyle=authoryear-comp]{biblatex}
```

Oltre a quelli predefiniti si possono usare molti altri stili: la bibliografia di questa guida, per esempio, è composta con lo stile `philosophy-modern`, presente in `TEX Live`. In Rete, infine, ce ne sono molti altri per comporre bibliografie conformi a precisi standard: si cerchi su  **CTAN**.

7.2.5 Comandi per le citazioni

Per citare un'opera nel documento si usano comandi particolari. Il più comune è `\cite`, che si usa come spiegato nel paragrafo 7.1 a pagina 121.

Il comando `\nocite`, invece, manda in bibliografia anche opere *non* citate nel documento (a patto, però, che figurino nel database):

- solo alcune, se dato nel corpo del documento al posto di `\cite` (come mostra il codice che produce la figura 12a a pagina 134);
- tutte, se dato prima o dopo `\printbibliography` (si veda più oltre) nella forma `\nocite{*}`.

Di seguito si mostrano gli stessi esempi d'uso di `\cite` con lo schema autore-anno compatto:

<code>\cite{eco:tesi} \\\</code>	Eco, 1977
<code>\cite[5]{eco:tesi} \\\</code>	Eco, 1977, p. 5
<code>\cite[5-9]{eco:tesi} \\\</code>	Eco, 1977, p. 5-9
<code>\cite[vedi][]{eco:tesi} \\\</code>	vedi Eco, 1977
<code>\cite[vedi][5]{eco:tesi}</code>	vedi Eco, 1977, p. 5

e con lo schema numerico compatto:

<code>\cite{eco:tesi} \</code>	[1]
<code>\cite[5]{eco:tesi} \</code>	[1, p. 5]
<code>\cite[5-9]{eco:tesi} \</code>	[1, pp. 5-9]
<code>\cite[vedi][]{eco:tesi} \</code>	[vedi 1]
<code>\cite[vedi][5]{eco:tesi}</code>	[vedi 1, p. 5]

(in modo del tutto analogo funzionano gli altri comandi: si facciano delle prove). Come si può osservare, biblatex produce automaticamente le abbreviazioni *p.* e *pp.* a seconda delle circostanze. (Si può ottenere anche la scrittura estesa *pagina* o *pagine*: si veda la documentazione del pacchetto.)

Oltre ai due comandi appena visti, biblatex ne definisce altri per citazioni ancora diverse:

- `\textcite` se la citazione è parte integrante del discorso: [Eco \[1977\]](#);
- `\parencite` racchiude la citazione fra parentesi: [\[Eco, 1977\]](#);
- `\footcite` mette la citazione in una nota, come qui¹;
- `\supercite` (solo per schemi numerici) mette la citazione in apice;
- `\fullcite` riporta nella citazione l'intero riferimento bibliografico nello stile impostato: Umberto Eco 1977 *Come si fa una tesi di laurea. Le materie umanistiche*, Bompiani, Milano.

Si noti che il tipo di parentesi dipende dall'opzione (*square*, in questa guida) assegnata a biblatex.

L'opzione `natbib` permette di mantenere una compatibilità quasi totale con i documenti la cui bibliografia è stata creata con il pacchetto `natbib`: la maggior parte dei nomi dei comandi per le citazioni da esso definiti, infatti, come `\citep` e `\citet`, sono conservati come alias.

Per citare singole parti di un riferimento ci sono `\citeauthor` e `\citeyear`:

<code>\citeauthor{eco:tesi} \</code>	Eco
<code>\citeyear{eco:tesi}</code>	1977

7.2.6 Generare e collocare la bibliografia nel documento

Il comando `\printbibliography` produce la sezione bibliografica con relativi titolo e testatina. Va dato *immediatamente dopo* le stesse sequenze di comandi già viste nel paragrafo 7.1 a pagina 121 per mandarne il relativo titolo nell'indice generale:

```
\cleardoublepage
%\phantomsection
\addcontentsline{toc}{chapter}{\bibname}
\printbibliography
```

se la classe è `book` o `report`, e

```
%\clearpage
%\phantomsection
\addcontentsline{toc}{section}{\refname}
\printbibliography
```

¹ [Eco, 1977](#).

se la classe è article. Anche per biblatex valgono le stesse osservazioni contenute in quel paragrafo, cui si rimanda.

Per generare effettivamente la bibliografia nel documento, infine, questa è la sequenza di composizione:

1. si compone il documento con L^AT_EX una prima volta;
2. si lancia il programma Biber premendo l'apposito pulsante dell'editor;
3. si compone il documento altre *due* volte con L^AT_EX.

7.2.7 Specialità

In questo paragrafo si presentano alcuni argomenti avanzati relativi a biblatex.

Campi speciali

I campi elencati di seguito non contengono dati stampabili, ma servono per scopi diversi e si possono applicare a *tutti* i record.

hyphenation Imposta la lingua del riferimento. Il valore deve essere il nome di una lingua nota a babel.

sortkey Imposta l'ordine alfabetico dei riferimenti: serve come chiave di ordinamento nei record privi dell'indicazione di autore o curatore.

keywords Un elenco di parole chiave separate da virgole, da usare come *filtro* sulle voci bibliografiche da far andare in bibliografia.

Campi omissibili

Alcuni dei campi indicati come obbligatori nel paragrafo 7.2.3 a pagina 125 in realtà non lo sono sempre. Nella bibliografia del proprio lavoro ci possono essere un libro anonimo oppure la raccolta dei lavori di una conferenza senza un curatore. Di regola questo non è un problema nella composizione della bibliografia, ma può esserlo nelle citazioni: uno schema di citazione autore-anno richiede *sempre* un campo author/editor e un campo date.

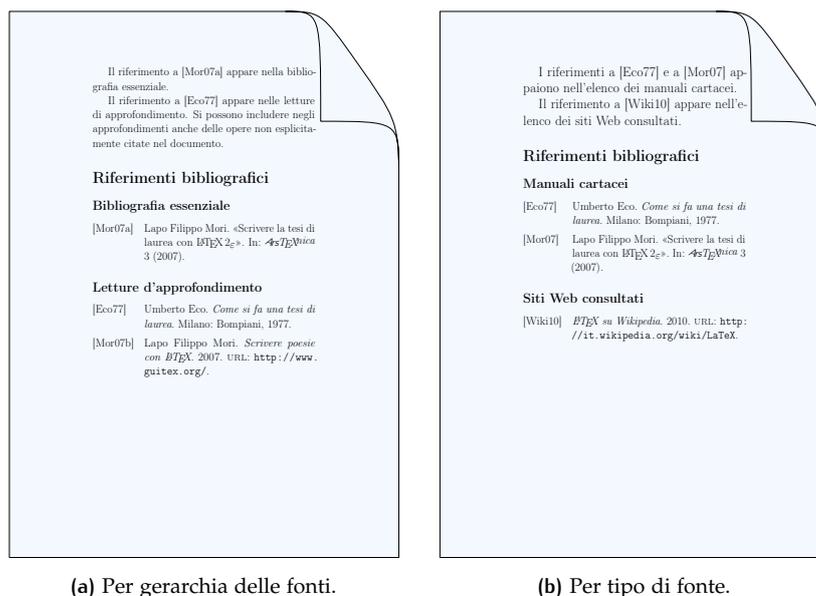
Se mancano i dati necessari per formare l'etichetta "regolare", si può sostituire ogni dato mancante con il campo label, da usare nei modi propri di ogni schema di citazione. Negli schemi autore-anno sostituisce il campo author/editor oppure il campo date nelle opere che ne sono prive. Negli schemi numerici, invece, non viene usato, perché in questo caso il formato delle citazioni è indipendente dai dati della bibliografia. Lo stesso accade negli schemi autore-titolo, perché il solo titolo basta di solito per formare una citazione univoca (di regola, ogni fonte ha un titolo).

Riferimenti finali cliccabili e bibliografia multilingue

Le seguenti opzioni di biblatex permettono di personalizzare la bibliografia.

backref Indica accanto a ciascun riferimento le pagine del documento in cui è citato.

hyperref Rende cliccabili le citazioni e i riferimenti (richiede il pacchetto omonimo).



(a) Per gerarchia delle fonti.

(b) Per tipo di fonte.

Figura 12: Bibliografie suddivise.

babel=hyphen Ordina a L^AT_EX di verificare per quale o per quali riferimenti si è specificata una lingua diversa da quella principale e di sillabarla con le regole della nuova lingua.

babel=other Dice a L^AT_EX di mettere il riferimento in un ambiente `otherlanguage` e di tradurre nella nuova lingua anche tutti i termini associati all'opera, come *curatore*, *volume*, eccetera.

L'esempio seguente mostra un record in cui si è specificata la lingua:

```
@book{lampport:latex,
  author      = {Lampport, Leslie},
  title       = {\LaTeX. A Document Preparation System},
  publisher    = {Addison-Wesley},
  date        = {1994},
  location     = {Reading (Massachusetts)},
  hyphenation = {english}
}
```

Suddividere la bibliografia

Talvolta è necessario suddividere la bibliografia in base a certi criteri. Per esempio, la si può ripartire in letteratura di base e di approfondimento, oppure elencare separatamente le risorse cartacee e quelle online, oppure ancora separare le opere citate esplicitamente nel documento da quelle che s'intendono semplicemente suggerire. Con `biblatex` tutto questo si può fare facilmente.

Si supponga di aver creato un database bibliografico `miabibliografia.bib` e si consideri il seguente codice:

```
\documentclass{article}
\usepackage[italian]{babel}
\usepackage[autostyle,italian=guillemets]{csquotes}
\usepackage[style=alphanumeric]{biblatex}
\usepackage{guit}
```

```

\addbibresource{miabibliografia.bib}

\DeclareBibliographyCategory{basi}
\DeclareBibliographyCategory{approf}

\addtocategory{basi}{mori:tesi}
\addtocategory{approf}{eco:tesi,mori:poesie}

\defbibheading{basi}{\subsection*{Bibliografia essenziale}}
\defbibheading{approf}{\subsection*{Letture d'approfondimento}}

\begin{document}

Il riferimento a-\cite{mori:tesi} appare nella bibliografia
essenziale.

Il riferimento a-\cite{eco:tesi} appare nelle letture
di approfondimento. Si possono includere negli approfondimenti
anche delle opere non esplicitamente citate nel documento.
\nocite{mori:poesie}

\section*{\refname}
\printbibliography[heading=basi,category=basi]
\printbibliography[heading=approf,category=approf]

\end{document}

```

Si tratta di un articolo con due bibliografie separate. Si osservi ora quanto segue.

- Il comando `\DeclareBibliographyCategory` nel preambolo definisce due *categorie bibliografiche*: una per la letteratura di base, l'altra per gli approfondimenti. Il comando `\addtocategory` assegna le opere a ciascuna categoria.
- Il comando `\defbibheading` definisce il titolo delle due bibliografie (*Bibliografia essenziale* e *Letture d'approfondimento*), che vengono composte esattamente come due sottoparagrafi (è la prassi, in un articolo). (Scrivendo un book o un report con bibliografia suddivisa, nel codice precedente `\subsection*` e `\section*{\refname}` vanno sostituiti con `\section*` e `\chapter*{\bibname}` rispettivamente.)
- L'argomento facoltativo di `\printbibliography` specifica le istruzioni di controllo: `heading=<nome>` imposta il titolo della bibliografia come specificato con `\defbibheading`, `category=<categoria>` filtra le voci da mettere in bibliografia, selezionando soltanto quelle che appartengono alla `<categoria>` specificata.

Al posto dei due `\printbibliography` con i relativi argomenti facoltativi si può scrivere un semplice

```
\bibbycategory
```

che equivale a dare un `\printbibliography` per ogni categoria nell'ordine in cui le si è dichiarate. Il risultato è mostrato nella figura 12a a fronte.

Un altro esempio. Si supponga di partire ancora dallo stesso database bibliografico appena considerato e si osservi il codice seguente:

```

\documentclass{article}
\usepackage[italian]{babel}
\usepackage[autostyle,italian=guillemets]{csquotes}

```

```

\usepackage[style=alphabetic]{biblatex}
\usepackage{guit}
\addbibresource{Bibliografia.bib}

\defbibheading{cartaceo}{\subsection*{Manuali cartacei}}
\defbibheading{web}{\subsection*{Siti Web consultati}}

\begin{document}

I riferimenti a~\cite{eco:tesi} e a~\cite{mori:tesi}
appaiono nell'elenco dei manuali cartacei.

Il riferimento a~\cite{wiki:latex} appare nell'elenco dei siti Web
consultati.

\section*{\refname}
\printbibliography[heading=cartaceo,nottype=online]
\printbibliography[heading=web,type=online]

\end{document}

```

Si tratta di un articolo con due bibliografie separate, una per i manuali cartacei e l'altra per i siti Web consultati (una "sitografia"). Si osservi che:

- il comando `\defbibheading` definisce il titolo delle due bibliografie (*Manuali cartacei* e *Siti Web consultati*);
- l'argomento facoltativo di `\printbibliography` filtra le voci come prima, per cui `type=<tipo>` manda in bibliografia soltanto quelle del *<tipo>* specificato, mentre `nottype=<tipo>` omette quelle il cui tipo *non* è *<tipo>*.

Il risultato è mostrato nella figura 12b a pagina 134.

Bibliografia di sezione

Il pacchetto biblatex permette di comporre facilmente sezioni bibliografiche anche alla fine di ogni capitolo (o paragrafo, se si tratta di un articolo) del documento. Se ne consultì la documentazione per sapere come fare.

8

INDICE ANALITICO

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

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

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

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

```
\index{voce}
```

la cui sintassi è mostrata nella tabella 48 nella pagina seguente, nel cui argomento in teoria si può scrivere ciò che si vuole.

Per produrre la sezione dell'indice analitico e mandarne il titolo nell'indice generale, *immediatamente prima* di `\end{document}` vanno date le stesse sequenze di comandi viste per la bibliografia nel paragrafo 7.1 a pagina 121:

```
\cleardoublepage
%\phantomsection
\addcontentsline{toc}{chapter}{\indexname}
\printindex
```

se la classe in uso è `book` o `report`, e

```
%\clearpage
%\phantomsection
\addcontentsline{toc}{section}{\indexname}
\printindex
```

se la classe è `article`. Il codice

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

Si possono inserire nell'indice analitico singole parole come `\emph{arte}\index{arte}`, oppure intere espressioni come questa `\index{interi espressioni come questa}`.

...

```
\addcontentsline{toc}{section}{\indexname}
\printindex
```

Tabella 48: Sintassi del comando `\index`.

Tipo di voce e codice	Risultato
Primaria <code>\index{Artisti}</code>	Artisti, 2
Sottolemma <code>\index{Artisti!Escher}</code>	Artisti, 2 Escher, 3
Voce in altro stile <code>\index{Gaudi@\textit{Gaudi}}</code>	<i>Gaudi</i> , 5
Pagina in altro stile <code>\index{Klimt textbf}</code>	Klimt, 7
Rimando <code>\index{Picasso see{Cubismo}}</code>	Picasso, <i>si veda</i> Cubismo
Intervallo di pagine <code>\index{Mirò } ... \index{Mirò }</code>	Mirò, 14–18

```
\end{document}
```

crea un indice analitico con le voci *arte* e *intere espressioni come questa*.

Per generare effettivamente l'indice analitico, questa è la sequenza di composizione da seguire:

1. si compone il documento con \LaTeX una prima volta;
2. si lancia il programma `MakeIndex` premendo l'apposito pulsante dell'editor;
3. si compone il documento altre *due* volte con \LaTeX .

Si ricorda infine il pacchetto `imakeidx`, con cui si possono ottenere automaticamente indici semplici e multipli *già alla prima composizione*.

Questo capitolo presenta listings, un potente pacchetto che gestisce la scrittura di codici in numerosi linguaggi di programmazione, controllandone molto finemente il formato.

Di qui in avanti si dà per caricato il pacchetto xcolor.

9.1 IMPOSTAZIONI

Il pacchetto si carica come al solito e di regola non c'è bisogno di preoccuparsi delle opzioni, anche se talvolta potrebbero essere utili (se ne veda la documentazione per i casi più importanti).

Il pacchetto s'impone *nel preambolo* con il comando `\lstset`, che agisce su *tutti i codici successivi* nello stesso gruppo e prende come argomento un elenco di opzioni del tipo $\langle \text{chiave} \rangle = \langle \text{valore} \rangle$. Per esempio, scrivendo

```
\lstset{basicstyle=\small\ttfamily}
```

si definisce uno stile generale (`basicstyle`) per *tutti* i codici del documento, che appariranno in corpo piccolo (`\small`, scelta raccomandata se in display) e con il font a spaziatura fissa predefinito (`\ttfamily`, a meno di non specificare diversamente). Per modificare *soltanto un* codice, si usino le opzioni di `\lstinline`, `\lstinputlisting` e `lstlisting` (per il loro significato si vedano i paragrafi successivi).

Si ricorda, infine, una limitazione piuttosto grave del pacchetto: listings non accetta i caratteri accentati e particolari della codifica Unicode inseriti direttamente dalla tastiera. In attesa che il problema venga risolto, si può usare l'opzione `literate`, la cui sintassi è la seguente:

```
literate={\langle testo originale \rangle}{\langle testo sostitutivo \rangle}{\langle larghezza \rangle}
```

dove:

- $\langle \text{testo originale} \rangle$ è, in questo caso, il carattere accentato o particolare inserito dalla tastiera;
- $\langle \text{testo sostitutivo} \rangle$ è il corrispondente carattere "costruito" con i comandi mostrati nella tabella 17 a pagina 54;
- $\langle \text{larghezza} \rangle$ è un numero intero che indica quanti caratteri il $\langle \text{carattere sostitutivo} \rangle$ occuperà nel testo composto.

L'opzione si usa come segue:

```
\lstset{literate={à}{\a}1{\ü}{\u}1{\dots}{\dots}1}
```

Si noti che in questo caso gli elementi *non* vanno separati con la virgola.

9.2 CODICI IN LINEA E IN DISPLAY

Come per citazioni e formule matematiche, si possono inserire anche i codici in linea e in display. Si noti che tutti i nomi di comandi e ambienti definiti dal pacchetto cominciano con `lst-`.

9.2.1 Codici in linea

Un codice in linea è un frammento di codice appartenente al flusso del discorso, come per esempio `var i : integer`. Per casi come questo, listings definisce il comando `\lstinline`, da usare così:

```
\lstinline[opzioni]{codice}
```

dove nelle *opzioni*, da trattare come al solito, si possono omettere i valori `=true` e come carattere delimitatore si può usare qualunque carattere *tranne una lettera o uno spazio*. Il *codice* si spiega da sé. Si ricorda che *non* si possono avere codici in linea riquadrati o con uno sfondo colorato.

Si noti che `\lstinline` eredita *tutte* le impostazioni dei vari `\lstset` che lo precedono nello stesso gruppo. Per esempio, dando nel preambolo l'opzione globale appena vista, anche `\lstinline` produrrà codici in corpo piccolo. Per averli sempre nel corpo corrente evitando di dare ogni volta `basicstyle=\normalsize`, basta introdurre tutti i codici in display (con le relative impostazioni) mediante gli ambienti personalizzati descritti nel paragrafo 9.6 a pagina 148.

9.2.2 Codici in display

Comandi standard

Un codice in display invece viene composto su righe a sé e bene in mostra sulla pagina, come per esempio:

```
1 for i:=maxint to 0 do
2 begin
3   { non far nulla }
4 end;
5 write('Benvenuto in Pascal.');
```

Per casi come questo listings definisce l'ambiente `lstlisting`, da usare come al solito.

Mettere in display un codice esterno

Il comando `\lstinputlisting` produce in display il contenuto di un file esterno (presente nella cartella di lavoro) e si usa come segue:

```
\lstinputlisting[opzioni]{nome del file con l'estensione}
```

Per esempio, le prime righe del file `arsclassica.sty` apparirebbero così:

<pre>\lstinputlisting[lastline=4]{arsclassica.sty}</pre>	<pre>1 %***** 2 % arclassica.sty 3 % 4 % copyright (C) 2008-2011 Lorenzo Pantieri</pre>
--	--

dove la chiave `lastline` dichiara l'ultima linea di codice da visualizzare (la quarta, in questo caso). Si ricordi che il percorso dei file caricati in questo modo *non* deve contenere spazi.

Mettere in display una porzione di codice

In coppia con l'analoga `firstline`, la chiave appena vista delimita la porzione di codice da riprodurre, come si vede nell'esempio seguente:

```
\begin{lstlisting}%
[firstline=2,lastline=5]
for i:=maxint to 0 do
begin
  { non far nulla }
end;

write('Benvenuto in Pascal.');
```

```
1 begin
2   { non far nulla }
3 end;
```

che mostra solo le righe dalla 2 alla 5 comprese (quelle vuote vengono ignorate).

Titolare un codice in display

Per dare un semplice titolo a un codice in display, si usa la chiave `title`, il cui valore va tra parentesi graffe:

```
\begin{lstlisting}[title={Un
semplice titolo.}]
for i:=maxint to 0 do
begin
  { non far nulla }
end;
\end{lstlisting}
```

Un semplice titolo.

```
1 for i:=maxint to 0 do
2 begin
3   { non far nulla }
4 end;
```

Assegnare etichetta e didascalia a un codice in display

Si possono assegnare etichetta e didascalia anche a un codice in display *non mobile* con le istruzioni consuete:

```
\dots qui finisce un capoverso.
\begin{lstlisting}[caption=%
{Un codice in display.},label=%
lst:display]
for i:=maxint to 0 do
begin
  { non far nulla }
end;
write('Benvenuto in Pascal.');
```

Il codice `\vref{lst:display}` è un esempio di codice in display con didascalia.

... qui finisce un capoverso.

Codice 1: Un codice in display.

```
1 for i:=maxint to 0 do
2 begin
3   { non far nulla }
4 end;
5 write('Benvenuto in Pascal.');
```

Il codice `1` è un esempio di codice in display con didascalia.

9.3 CODICI MOBILI

Comandi standard

Si può trasformare facilmente un codice in display in oggetto mobile da richiamare eventualmente con un riferimento incrociato. Per esempio, il codice `2` nella pagina successiva è stato ottenuto con le istruzioni:

```
\begin{lstlisting}[float=tb,caption={Esempio di codice mobile.},%
captionpos=t,label=lst:esempio]
for i:=maxint to 0 do
begin
  { non far nulla }
```

Codice 2: Esempio di codice mobile.

```

1 for i:=maxint to 0 do
2 begin
3     { non far nulla }
4 end;
5 write('Benvenuto in Pascal.');
```

```

end;
write('Benvenuto in Pascal.');
```

```

\end{lstlisting}
```

Si noti quanto segue:

- la chiave `float`, la cui sintassi generale è

```
float=[*]<preferenze di collocazione>
```

accetta le stesse *<preferenze di collocazione>* degli oggetti mobili viste nel paragrafo 6.2.2 a pagina 93 e l'asterisco permette di ottenere un codice a tutta pagina in un documento a più colonne;

- `caption` e `label` si comportano come al solito;
- la chiave `captionpos`, che accetta solo i valori `t` o `b`, permette di avere la didascalia sopra o sotto il codice, rispettivamente.

Indice dei codici e personalizzazioni

Il comando `\lstlistoflistings`, analogo ai comandi per gli elenchi di figure e tabelle, produce l'elenco dei codici contenuti nel documento nel punto in cui viene dato.

Si può specificare una didascalia breve come segue:

```
caption={[<didascalia breve>]<didascalia>}
```

Si notino le graffe di raggruppamento. Se non si vuole mandare il codice nell'elenco dei codici, l'opzione è `no1o1`.

Per avere *Codice* anziché *Listing* ed *Elenco dei codici* anziché *List of Listings*, basta scrivere nel *preambolo* (dopo aver caricato `babel` con l'opzione `italian`):

```
\addto\captionsitalian{%
\renewcommand{\lstlistingname}{Codice}}
```

e

```
\addto\captionsitalian{%
\renewcommand{\lstlistlistingname}{Elenco dei codici}}
```

9.4 GESTIRE I DIVERSI LINGUAGGI

Il pacchetto riconosce un elevato numero di linguaggi di programmazione, alcuni dei quali (dialetti compresi) sono mostrati nella tabella 49 nella pagina successiva (il loro elenco completo si trova nella documentazione). Per indicare a `listings` il linguaggio da usare la chiave è `language`, con la seguente sintassi:

```
language={[<dialetto>]<linguaggio>}
```

Tabella 49: Alcuni linguaggi di programmazione riconosciuti da listings. I dialetti sottolineati sono predefiniti.

Algol (60, <u>68</u>)	Ada (<u>2005</u> , 83, 95)
Basic (Visual)	C (<u>ANSI</u> , Objective)
C++ (<u>ANSI</u> , GNU, <u>ISO</u> , Visual)	Cobol (1974, <u>1985</u>)
Delphi	Fortran (77, 90, <u>95</u>)
HTML	Java
Lisp	Mathematica (1.0, 3.0, <u>5.2</u>)
Matlab	METAPOST
Modula-2	Pascal (Borland6, <u>Standard</u> , XSC)
Perl	PHP
PostScript	Prolog
R	SQL
TeX (LaTeX, <u>plain</u>)	XML

Omettendo [*(dialetto)*], viene caricato il dialetto predefinito. Per esempio:

```
\lstset{language=[77]Fortran}
```

seleziona il Fortran 77 e

```
\lstset{language=Pascal}
```

seleziona il Pascal Standard.

9.5 PERSONALIZZARE L'ASPETTO DEI CODICI

Il pacchetto listings permette di regolare molto finemente il formato del codice. Negli esempi di codice proposti fin qui, tutti in Pascal, le parole chiave erano in blu, i commenti in verde e le stringhe in rosso.

Si può ottenere un risultato diverso come il seguente:

<pre>\begin{lstlisting} for i:=maxint to 0 do begin { non far nulla } end; write('Benvenuto in Pascal.');</pre>	<pre>1 for i:=maxint to 0 do 2 begin 3 { non far nulla } 4 end; 5 write('Benvenuto_in_Pascal.');</pre>
---	--

impostando il pacchetto così:

```
\lstset{basicstyle=\small\ttfamily,%
keywordstyle=\color{black}\bfseries,%
commentstyle=\color{gray},%
stringstyle=\color{black},%
showstringspaces}
```

Si noti che:

- `keywordstyle` imposta lo stile delle parole chiave (qui in nero sia nel colore che nello stile);
- `commentstyle` imposta l'aspetto dei commenti (qui in grigio);
- `stringstyle` imposta l'aspetto delle stringhe (qui di colore nero);
- `showstringspaces` evidenzia gli spazi nelle stringhe con `_`.

In generale, per i codici si consiglia di seguire il criterio sempre valido della sobrietà, tenendo presente che lo stile da preferire dipende anche dal tipo di documento in lavorazione: una presentazione, per esempio, spesso richiede un impatto visivo maggiore di quello adatto a un libro.

Di seguito si presenta una panoramica di strumenti utili per personalizzare i codici. Per maggiori dettagli, si rinvia alla ricca documentazione del pacchetto.

9.5.1 Numerare le righe

Si possono numerare le righe di un codice in display e personalizzare ogni elemento della numerazione, come nell'esempio seguente:

```
\lstset{numbers=left,
  numberstyle=\tiny,stepnumber=2}

\begin{lstlisting}
for i:=maxint to 0 do
begin
  { non far nulla }
end;
write('Benvenuto in Pascal.');
```

```
for i:=maxint to 0 do
begin
  { non far nulla }
end;
write('Benvenuto in Pascal.');
```

Si noti che:

- `numbers` imposta la posizione dei numeri (qui a sinistra);
- `numberstyle` imposta il corpo dei numeri (qui molto piccolo);
- `stepnumber` imposta l'intervallo di numerazione delle righe (qui ogni due).

Si può interrompere un codice e riprenderlo successivamente mantenendo la numerazione corretta, come mostra l'esempio seguente:

```
\begin{lstlisting}%
[firstnumber=100]
for i:=maxint to 0 do
begin
  { non far nulla }
end;

\end{lstlisting}
Riprendiamo il codice:
\begin{lstlisting}%
[firstnumber=last]
write('Benvenuto in Pascal.');
```

```
100 for i:=maxint to 0 do
101 begin
102   { non far nulla }
103 end;
```

Riprendiamo il codice:

```
105 write('Benvenuto in Pascal.');
```

La chiave `firstnumber` dichiara la prima riga del codice, e può assumere un valore numerico o posizionale. Nella prima parte dell'esempio è impostata a 100, come si vede a destra. Nella seconda assume il valore `last` (corrispondente alla riga 104, contata ma non visualizzata perché vuota), utile per continuare la numerazione senza dover contare a mano le righe.

Scrivendo una volta per tutte nel preambolo

```
\lstset{firstnumber=last}
```

si ottengono le righe di tutti i codici del documento numerate consecutivamente (a meno di non specificare diversamente per codici particolari).

In alternativa, si può assegnare un nome a un codice (distinguendo maiuscole e minuscole) con la chiave `name`: parti diverse dello stesso codice condividono lo stesso contatore di riga. La si vede all'opera nell'esempio seguente:

<pre>\begin{lstlisting}[name=Test] for i:=maxint to 0 do begin { non far nulla } end; \end{lstlisting} Riprendiamo il codice: \begin{lstlisting}[name=Test] write('Benvenuto in Pascal.');</pre>	<pre>1 for i:=maxint to 0 do 2 begin 3 { non far nulla } 4 end;</pre>
<pre>\end{lstlisting}</pre>	<p>Riprendiamo il codice:</p> <pre>5 write('Benvenuto in Pascal.');</pre>

La prima riga del successivo codice `Test` sarà la numero 6, indipendentemente dalla presenza di altri codici in mezzo.

9.5.2 Rientri

Tabulazioni e spazi

I rientri (per mezzo di tabulazioni o spazi) servono a rendere il codice più leggibile. Il pacchetto assume che le tabulazioni si arrestino alle colonne 9, 17, 25, 33, ..., valori che dipendono dalla chiave `tabsize`, impostata a 8 di default. Ponendo il valore della chiave a n , le interruzioni si arresteranno alle colonne $n + 1$, $2n + 1$, $3n + 1$, e così via.

L'esempio seguente mostra il comportamento delle tabulazioni:

<pre>\lstset{tabsize=2} \begin{lstlisting} 123456789 { una tabulazione } { due tab } 123 { 123 + due tab }</pre>	<pre>1 123456789 2 { una tabulazione } 3 { due tab } 4 123 { 123 + due tab }</pre>
<pre>\end{lstlisting}</pre>	

Si noti che anche se \LaTeX è impostato con `tabsize=4`, la tabulazione risultante corrisponde al valore 2 impostato.

Rendere visibili tabulazioni e spazi

Si possono rendere visibili i segni di tabulazione e gli spazi con le chiavi `showtabs` e `showspaces` rispettivamente.

<pre>\lstset{showspaces, showtabs, tabsize=8, tab=\rightarrowfill} \begin{lstlisting} for i:=maxint to 0 do begin { non far nulla } end; \end{lstlisting}</pre>	<pre>1 _____for_i:=maxint_to_0_do 2 _____begin 3 _____>{non_far_nulla_} 4 _____end;</pre>
--	--

Si noti che la seconda chiave da sola trasforma le tabulazioni in spazi.

Tabella 50: Valori di frame e loro comportamento.

Valore	Che cosa produce?
topline	Linea sopra il codice
bottomline	Linea sotto il codice
leftline	Linea a sinistra del codice
lines	Linee sopra e sotto il codice
single	Riquadro singolo
shadowbox	Riquadro ombreggiato
none	Nessun riquadro
t/T	Linea semplice/doppia sopra il codice
b/B	Linea semplice/doppia sotto il codice
r/R	Linea semplice/doppia a destra del codice
l/L	Linea semplice/doppia a sinistra del codice

9.5.3 Riquadri

Per racchiudere un codice in un riquadro c'è la chiave `frame`, che può assumere i valori mostrati nella tabella 50. Di seguito se ne mostrano due esempi.

```
\begin{lstlisting}[frame=lines]
for i:=maxint to 0 do
begin
  { non far nulla }
end;
\end{lstlisting}
```

```
1 for i:=maxint to 0 do
2 begin
3   { non far nulla }
4 end;
```

```
\begin{lstlisting}[frame=trBL]
for i:=maxint to 0 do
begin
  { non far nulla }
end;
\end{lstlisting}
```

```
1 for i:=maxint to 0 do
2 begin
3   { non far nulla }
4 end;
```

Si può scegliere anche il profilo degli angoli del riquadro, arrotondati (t) o diritti (f), con la chiave `framround`, che prende come valore un elenco di quattro caratteri, scelti fra le due lettere appena viste. Il prossimo esempio la mostra all'opera:

```
\lstset{framround=fttt}

\begin{lstlisting}[frame=trBL]
for i:=maxint to 0 do
begin
  { non far nulla }
end;
\end{lstlisting}
```

```
1 for i:=maxint to 0 do
2 begin
3   { non far nulla }
4 end;
```

Si noti che il primo carattere del valore è relativo all'angolo in alto a destra, e gli altri proseguono in senso orario.

9.5.4 Sfondi colorati

Per mettere dietro al codice uno sfondo colorato c'è `backgroundcolor`, da usare come segue:

```
backgroundcolor=\color{<colore>}
```

dove il $\langle\text{colore}\rangle$ va scelto tra quelli previsti dal pacchetto xcolor. Di seguito se ne mostra un esempio:

```
\lstset{backgroundcolor=%
\color{Apricot}}
```

```
\begin{lstlisting}[frame=single]
for i:=maxint to 0 do
begin
  { non far nulla }
end;
\end{lstlisting}
```

```
1 for i:=maxint to 0 do
2 begin
3   { non far nulla }
4 end;
```

9.5.5 Evidenziare parole

Il pacchetto listings evidenzia automaticamente le parole chiave di un codice. Per evidenziarne altre *a mano*, ci sono le chiavi `emph` e `emphstyle`, che si vedono all'opera negli esempi seguenti:

```
\lstset{emph={square,root},%
emphstyle=\bfseries}
```

```
\begin{lstlisting}
for i:=maxint to 0 do
begin
  j:=square(root(i));
end;
\end{lstlisting}
```

```
1 for i:=maxint to 0 do
2 begin
3   j:=square(root(i));
4 end;
```

```
\lstset{emph={square},%
emphstyle=\color{Green},%
emph={[2]root},%
emphstyle={[2]\color{Red}}}
```

```
\begin{lstlisting}
for i:=maxint to 0 do
begin
  j:=square(root(i));
end;
\end{lstlisting}
```

```
1 for i:=maxint to 0 do
2 begin
3   j:=square(root(i));
4 end;
```

Si noti che:

- `emph` prende come argomento l'elenco delle parole da evidenziare, separate con la virgola se più d'una;
- `emphstyle` ne regola l'aspetto in quanto a corpo e colore;
- l'argomento facoltativo delle chiavi permette di selezionare ulteriormente le parole da evidenziare con stili ancora diversi.

9.5.6 Indicizzare parole di un codice

Per mandare una parola del codice nell'indice analitico c'è la chiave `index`, che prende come argomento l'elenco delle parole in questione, da separare con la virgola se più d'una:

```

\lstset{index={square,root}}

\begin{lstlisting}
for i:=maxint to 0 do
begin
  j:=square(root(i));
end;
\end{lstlisting}

```

```

1 for i:=maxint to 0 do
2 begin
3   j:=square(root(i));
4 end;

```

9.5.7 Allineare le colonne

Oltre a colori, rientri e riquadri, uno degli aspetti tipografici più evidenti nella composizione dei codici è l'allineamento delle colonne, che di regola va mantenuto. Si noti che se l'allineamento riveste anche una funzione sintattica bisogna usare un font a spaziatura fissa; in caso contrario, si usi il font desiderato.

A tal fine, si danno di seguito alcuni consigli:

- si usi *sempre* l'opzione `columns=fullflexible`, per mantenere la larghezza naturale dei caratteri;
- non si usi *mai* l'opzione `columns=fixed`, con cui listings cerca di allineare le colonne modificando gli spazi interparola, con risultati tipografici decisamente discutibili;
- per evitare il comportamento appena descritto, si usi *sempre* l'opzione `keepspaces=true`.

9.6 TECNICHE AVANZATE

Ambienti personali

Il pacchetto listings permette di definire ambienti personali con il comando `\lstnewenvironment`, la cui sintassi è del tutto analoga a quella del comando standard `\newenvironment` già vista nel paragrafo 11.1.2 a pagina 174:

```

\lstnewenvironment{<nome>}[<numero di argomenti>]%
  {<comandi di apertura>}{<comandi di chiusura>}

```

Scrivendo nel preambolo (si noti l'argomento vuoto):

```

\lstnewenvironment{pascal}{\lstset{language=pascal}}{}

```

si potrà poi usare l'ambiente pascal nel modo seguente:

```

\begin{pascal}
for i:=maxint to 0 do
begin
  { non far nulla }
end;
write('Benvenuto in Pascal. ');
\end{pascal}

```

```

1 for i:=maxint to 0 do
2 begin
3   { non far nulla }
4 end;
5 write('Benvenuto in Pascal. ');

```

Linguaggi personali

Il pacchetto, infine, permette di definire *ex novo* linguaggi personali, specificando in ogni aspetto il formato di parole chiave, commenti, stringhe e delimitatori (se ne veda la documentazione).

10 | GRAFICI

Questo capitolo, basato su [De Marco e Giacomelli, 2011], cui si rimanda per ogni approfondimento, presenta `pgfplots`, un pacchetto per la rappresentazione grafica di dati derivato da `PGF/TikZ`. Si mostrerà come impostare un sistema di riferimento, come scrivere le istruzioni per creare grafici di funzione e d'altro tipo, come realizzare diagrammi a barre e come personalizzare l'aspetto dei vari elementi del disegno. Non si prenderà in considerazione, invece, la costruzione di grafici da zero, per la quale si rimanda al pacchetto `TikZ` e alla sua ponderosa documentazione.

10.1 GRAFICI E TIPOGRAFIA

La rappresentazione di dati numerici tramite grafici di vario tipo è una parte consistente e indispensabile della comunicazione tecnico-scientifica, perché molto spesso permette di esporre concetti matematici e fenomeni fisici in modo molto più semplice e intuitivo di quanto farebbero formule da sole o elenchi di numeri o tabelle.

Come fare per inserire un grafico in un documento \LaTeX ? Una via potrebbe essere quella di includerlo come file esterno prodotto con un programma specializzato: Mathematica, MATLAB e Octave, per citarne alcuni, generano grafici molto sofisticati e li possono esportare nel formato PDF accettato da \LaTeX .

Gli inconvenienti, però, non tardano a presentarsi, perché un disegno importato, in generale:

- usa font diversi da quelli del proprio documento;
- contiene simboli matematici ed elementi grafici che male si adattano allo stile scelto;
- peggio: le formule matematiche, se presenti, appaiono completamente diverse;
- presenta linee o troppo grosse o troppo sottili.

10.2 IL PACCHETTO `pgfplots`

Il pacchetto `pgfplots`, compreso in tutte le distribuzioni complete di \LaTeX , permette di comporre grafici coerenti con le impostazioni tipografiche del documento in lavorazione scrivendone le istruzioni *direttamente* nel testo sorgente e assicurando la più alta qualità tipica di \LaTeX .

Con `pgfplots` si possono tracciare curve e superfici di qualunque tipo, in due e tre dimensioni, creare diagrammi a barre e altri grafici particolari, aggiungervi etichette, legende, titoli e personalizzare ogni elemento del disegno. Inoltre, `pgfplots` può eseguire i calcoli necessari sfruttando le stesse capacità di \LaTeX senza appoggiarsi a strumenti esterni.

Il pacchetto carica automaticamente `TikZ` (permettendo eventualmente di usarne i comandi) e `xcolor`, *dopo* il quale (se già presente nel preambolo) va

Tabella 51: Sistemi di riferimento disponibili in pgfplots e librerie richieste.

Sistema di riferimento	Ambiente	Libreria richiesta
Cartesiano	axis	
Cartesiano logaritmico	loglogaxis	
Ascissa logaritmica	semilogxaxis	
Ordinata logaritmica	semilogyaxis	
Coordinate polari	polaraxis	polar
Diagramma ternario	ternaryaxis	ternary
Carta di Smith	smithchart	smithchart

sempre caricato. Eventuali opzioni valide per tutti i grafici del documento si possono mettere nell'argomento di `\pgfplotsset` nel *preambolo*:

```
\usepackage{pgfplots}
\pgfplotsset{/pgf/number_format/use_comma,compat=newest, <altre opzioni>}
```

Si raccomanda di scrivere le opzioni *rispettando sempre gli eventuali spazi* (qui evidenziati con `_`), e in particolare le due nell'esempio precedente:

- la prima imposta la virgola come separatore decimale (le istruzioni nel sorgente, invece, richiedono il punto);
- la seconda assicura che si usino le caratteristiche della versione più recente del pacchetto.

Si noti, infine, che i grafici prodotti da pgfplots sono oggetti *in testo*, con tutti i possibili inconvenienti del caso. Nulla vieta però di renderli *mobili* semplicemente scrivendone il codice in un ambiente `figure` come si mostra di seguito:

```
\begin{figure}
\centering
\begin{tikzpicture}
...
\end{tikzpicture}
\caption{...}
\label{fig:...}
\end{figure}
```

10.2.1 Impostare il sistema di riferimento: l'ambiente `axis`

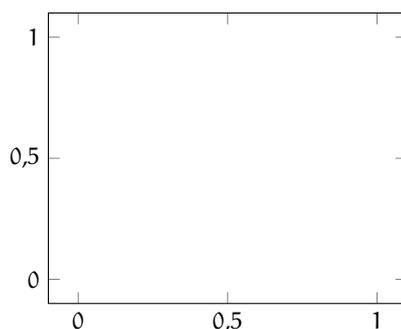
L'ambiente fondamentale di pgfplots è `axis`, da inserire a propria volta nell'ambiente `tikzpicture` (definito da `TikZ`) con la sintassi:

```
\begin{tikzpicture}
\begin{axis}[<opzioni>]
<istruzioni di pgfplots o TikZ>
\end{axis}
\end{tikzpicture}
```

Si noti che le *<opzioni>*, se presenti, agiranno su *tutti* i grafici inseriti in quell'ambiente `axis` (e solo su quelli).

Eccone un esempio davvero minimo:

```
\begin{tikzpicture}
\begin{axis}
\end{axis}
\end{tikzpicture}
```



Si noti che:

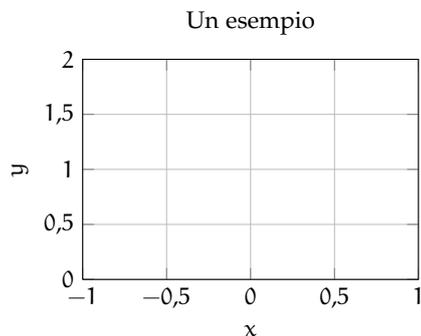
- `axis` definisce un sistema di riferimento cartesiano ortogonale visualizzandone parte del primo quadrante (e una piccola porzione degli altri tre) come un riquadro;
- i lati del riquadro portano tacche di marcatura distanti tra loro un certo passo, calcolato automaticamente in base a parametri interni;
- per esigenze tipografiche i disegni di questo capitolo hanno larghezza fissa: riproducendoli, si potrebbero avere risultati diversi (ma non scorretti).

Gli altri ambienti elencati nella tabella 51 a fronte producono i sistemi di riferimento mostrati nella figura 13 nella pagina successiva e nel paragrafo 10.6 a pagina 169, nel quale si spiega anche come caricare le librerie richieste.

Si può personalizzare il risultato predefinito passando ad `axis` opportune opzioni nella notazione $\langle \text{chiave} \rangle = \langle \text{valore} \rangle$ o anche solo $\langle \text{chiave} \rangle$ (i valori `=true` si possono omettere).

Il prossimo è un esempio con qualche opzione:

```
\begin{tikzpicture}
\begin{axis} [xmin=-1,xmax=1,
ymin=0,ymax=2,grid=major,
xlabel=$x$,ylabel=$y$,
title={Un esempio},
width=6cm,height=4.5cm]
\end{axis}
\end{tikzpicture}
```

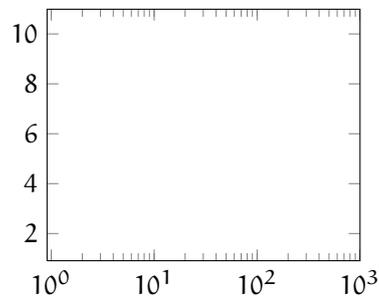


Si noti quanto segue.

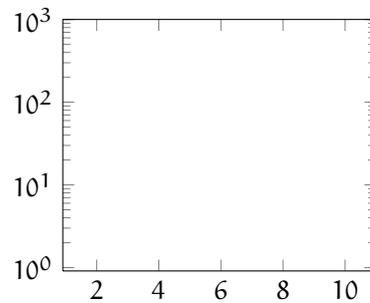
- `xmin` e `ymax` fissano rispettivamente il valore minimo e massimo delle ascisse; analoghe chiavi si useranno per le ordinate.
- `grid=major` visualizza una griglia agganciata alle tacche di marcatura degli assi per leggere più facilmente il grafico.
- `xlabel` e `ylabel` producono le etichette degli assi (non obbligatorie), qui x e y . Quest'ultima per impostazione predefinita è ruotata di 90° in senso antiorario e centrata verticalmente: per averla diritta, come mostra l'esempio seguente, basta scrivere tra le opzioni di `axis`:

```
 $\langle \text{nome dell'etichetta} \rangle \text{label style} = \{ \text{rotate} = -90 \}$ 
```

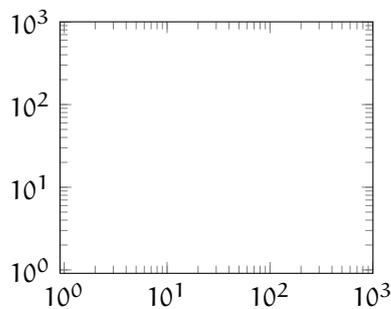
Se il valore di un'etichetta contiene caratteri particolari, va racchiuso tra parentesi graffe.



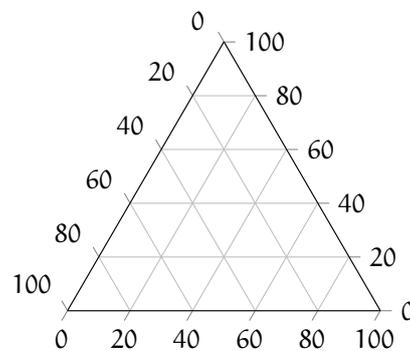
(a) Piano con ascissa logaritmica.



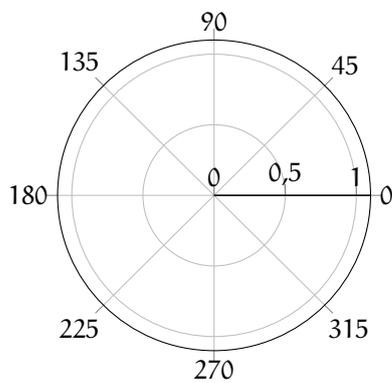
(b) Piano con ordinata logaritmica.



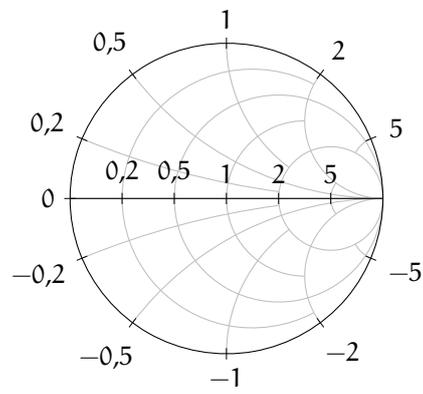
(c) Piano logaritmico.



(d) Diagramma ternario.



(e) Sistema di coordinate polari.



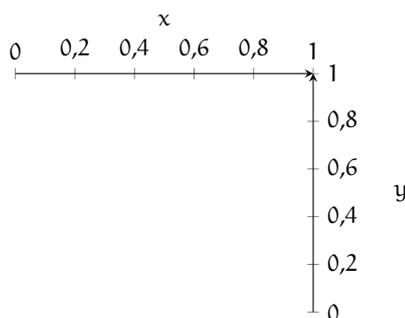
(f) Carta di Smith.

Figura 13: Sistemi di riferimento predefiniti di pgplots (tranne quello prodotto da axis).

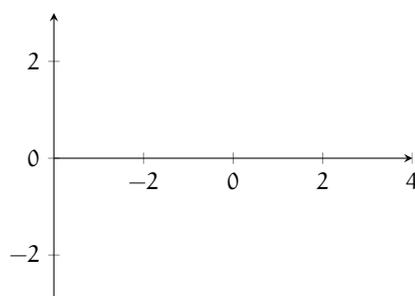
- `title` produce un titolo sopra il disegno (ma se il grafico è mobile, lo si metta nell'argomento di `\caption`).
- `width` e `height`, esprimibili come al solito, impostano rispettivamente larghezza e altezza dell'intero disegno, etichette e titoli *compresi*. Per riferire le dimensioni al solo riquadro, invece, si aggiunga l'opzione `scale only axis`.

Di seguito si mostrano alcuni esempi in cui si è modificata la posizione degli assi. Si noti che dichiararli esplicitamente attiva lo stile tradizionale ed elimina il riquadro.

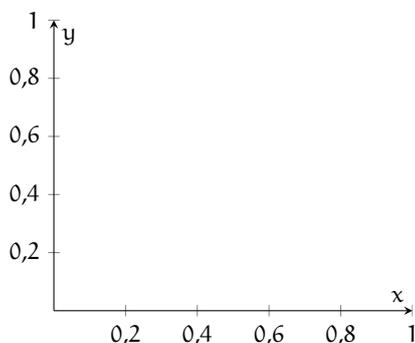
```
\begin{tikzpicture}
\begin{axis}
[axis x line=top,
axis y line=right,
xlabel=$x$,ylabel=$y$,
ylabel style={rotate=-90}]
\end{axis}
\end{tikzpicture}
```



```
\begin{tikzpicture}
\begin{axis}
[xmin=-4,xmax=4,ymin=-3,ymax=3,
axis x line=middle,
axis y line=left]
\end{axis}
\end{tikzpicture}
```



```
\begin{tikzpicture}
\begin{axis}
[axis lines=middle,
xlabel=$x$,ylabel=$y$]
\end{axis}
\end{tikzpicture}
```



Si noti quanto segue.

- `axis x line` regola la posizione delle ascisse: accetta i valori `bottom`, `middle` e `top`, che ne impongono il passaggio rispettivamente per $y = y_{\min}$, $y = 0$ e $y = y_{\max}$. (Quando però il grafico è tutto sotto o sopra l'asse x , il valore `middle` corrisponde a `top` o `bottom` rispettivamente.)
- `axis y line` regola la posizione delle ordinate: accetta i valori `left`, `middle` e `right`, che ne impongono il passaggio per $x = x_{\min}$, $x = 0$ e $x = x_{\max}$, rispettivamente. (Quando però il grafico è tutto a sinistra o a destra dell'asse y , il valore `middle` corrisponde a `right` o `left` rispettivamente.)

Tabella 52: Alcune funzioni e costanti matematiche predefinite da pgfplots.

Funzione	Significato	Funzione	Significato
<code>sqrt</code>	radice quadrata	<code>cot</code>	cotangente
<code>exp</code>	esponenziale	<code>asin</code>	arcoseno
<code>ln</code>	logaritmo naturale	<code>acos</code>	arcocoseno
<code>log10</code>	logaritmo decimale	<code>atan</code>	arcotangente
<code>sin</code>	seno	<code>sinh</code>	seno iperbolico
<code>cos</code>	coseno	<code>cosh</code>	coseno iperbolico
<code>tan</code>	tangente	<code>tanh</code>	tangente iperbolica
<code>sec</code>	secante	<code>e</code>	numero di Nepero
<code>cosec</code>	cosecante	<code>pi</code>	π

- `axis lines` imposta contemporaneamente entrambi gli assi con il valore scelto, di solito `middle` (se non li si desidera visualizzare, il valore da dare è `none`) e raddrizza automaticamente l'etichetta delle ordinate.

La documentazione del pacchetto, consultabile con `texdoc pgfplots`, spiega come personalizzare la posizione delle etichette.

10.2.2 Disegnare il grafico: i comandi `\addplot` e `\addplot3`

Il comando `\addplot`, dato in uno degli ambienti di `pgfplots` e ripetibile per ogni grafico da aggiungere nel sistema di riferimento, disegna un grafico nel piano. La sintassi generale è la seguente:

```
\addplot [opzioni]  
{superistruzione}  
{istruzioni};
```

Si noti che:

- *opzioni*, da esprimere come al solito, sono le opzioni che definiscono l'aspetto del grafico;
- la *superistruzione* (`coordinates`, `file` o `table`) va indicata nei casi spiegati tra poco;
- *istruzioni* sono le istruzioni per ottenere il grafico vero e proprio;
- il punto e virgola immediatamente dopo l'argomento di `\addplot` è obbligatorio;
- si possono separare i vari elementi del codice uno spazio bianco per rendere il sorgente più leggibile.

Le *istruzioni* possono essere dei tre tipi descritti di seguito.

- Un'espressione matematica, che verrà valutata per un opportuno numero di punti del dominio di definizione, con il codice

```
\begin{axis}[...]  
\addplot [...]  
{espressione matematica};  
\end{axis}
```

- Una sequenza di coppie di valori (corrispondenti a coordinate di punti del piano), con il codice

```

\begin{axis}[\langle...\rangle]
\addplot [\langle...\rangle] coordinates
{(x1, y1) (x2, y2)
... (xn, yn)};
\end{axis}

```

dove `coordinates` ordina a `pgfplots` di disegnare il grafico usando le coordinate scritte nell'argomento.

- Una sequenza di coppie di valori separati da almeno uno spazio e disposte su più righe, contenuta in un file esterno prodotto con uno dei programmi nominati nel paragrafo 10.1 a pagina 149 e sistemato nella cartella di lavoro. Il codice generale è il seguente:

```

\begin{axis}[\langle...\rangle]
\addplot [\langle...\rangle]
file {\langlenome del file di dati con l'estensione\rangle};
\end{axis}

```

dove `file` ordina a `pgfplots` di usare il file indicato (registrato con una delle estensioni accettate dal pacchetto). In alternativa si può usare `table`, una variante di `file` personalizzabile (si veda la documentazione del pacchetto).

Con la stessa sintassi, il comando `\addplot3` permette di disegnare un grafico nello spazio. Nei casi in cui l'istruzione non sia un'espressione matematica, le coordinate vanno espresse come terne anziché coppie di valori.

10.3 FUNZIONI ESPRESSE ANALITICAMENTE

La tabella 52 a fronte mostra le più importanti funzioni e costanti matematiche predefinite da `pgfplots`. Di seguito se ne presentano alcune realizzazioni.

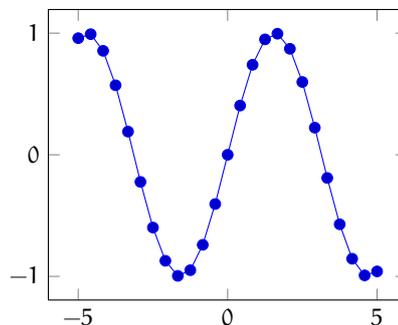
10.3.1 Funzioni reali d'una variabile reale

Un semplice esempio senza opzioni per cominciare.

```

\begin{tikzpicture}
\begin{axis}
\addplot {sin(deg(x))};
\end{axis}
\end{tikzpicture}

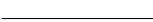
```



Si noti che:

- il risultato predefinito è una curva blu contrassegnata da marcatori circolari (per avere solo i marcatori, ma in nero, si passi a `\addplot` la chiave `only marks`);

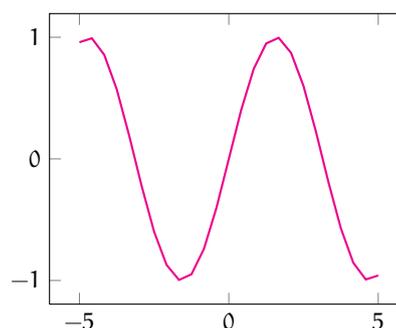
Tabella 53: Spessori di linea disponibili in pgfplots.

Chiave	Risultato	Chiave	Risultato
ultra thin		thick	
very thin		very thick	
thin		ultra thick	
semithick			

- `deg` trasforma in gradi il proprio argomento, da mettere fra parentesi *tonde* (pgfplots assume che l'argomento delle funzioni trigonometriche sia espresso in gradi e non in radianti).

Ora un esempio con qualche personalizzazione:

```
\begin{tikzpicture}
\begin{axis}
\addplot [thick,color=magenta]
{sin(deg(x))};
\end{axis}
\end{tikzpicture}
```



Si noti che:

- non ci sono marcatori, perché le opzioni di `\addplot` sostituiscono le impostazioni predefinite (per mantenerle aggiungendovi nuove opzioni o per ridefinire localmente un'opzione, il comando è `\addplot+`; in questo caso, per eliminare del tutto i marcatori la chiave è `no marks`);
- `thick` imposta lo spessore della curva, da scegliere tra quelli mostrati nella tabella 53;
- `color=magenta` (ma basta scrivere `magenta`) imposta il colore desiderato per la curva, da scegliere tra quelli elencati nella tabella 54 a fronte (se non bastassero, si possono usare le tavolozze predefinite di `xcolor` o definirne uno personale regolandone a mano le componenti).

La curva dell'esempio precedente è leggermente spigolosa, perché pgfplots disegna le curve approssimandole alla spezzata che unisce un opportuno campione di punti. Se questi sono sufficientemente vicini, si percepisce la spezzata come una curva "regolare", tracciata con precisione e qualità; ma se sono insufficienti, il risultato è inaccettabile, come nell'esempio seguente:

```
\begin{tikzpicture}
\begin{axis}
\addplot [thick,blue]
{sin(5*deg(x))};
\end{axis}
\end{tikzpicture}
```

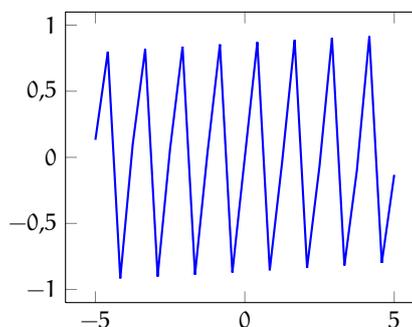
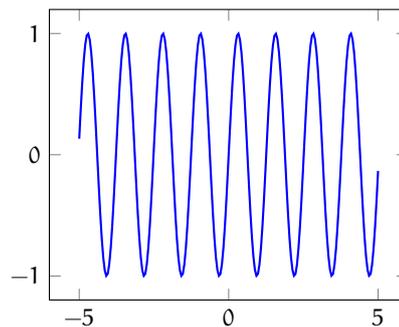


Tabella 54: Colori disponibili in pgfplots.

Chiave	Risultato	Chiave	Risultato	Chiave	Risultato
red		brown		orange	
green		lime		olive	
blue		cyan		teal	
magenta		purple		violet	
gray		darkgray		lightgray	
yellow		black		pink	
white					

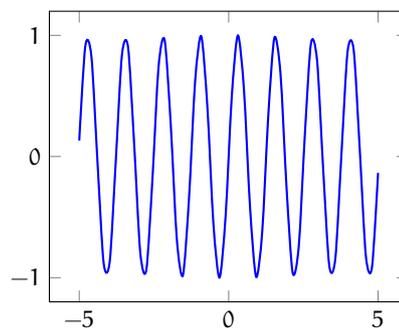
Aumentando il valore della chiave `samples`, che gestisce il numero di punti campionati (il suo valore predefinito è 25), si risolve il problema:

```
\begin{tikzpicture}
\begin{axis}
\addplot
[samples=200,thick,blue]
{sin(5*deg(x))};
\end{axis}
\end{tikzpicture}
```



Si può anche usare la chiave `smooth`, da sola o insieme alla precedente: essa ordina al programma di tracciare il grafico con curve di Bézier (cubiche raccordate con continuità di tangente e concavità) anziché con segmenti.

```
\begin{tikzpicture}
\begin{axis}
\addplot
[samples=50,smooth,thick,blue]
{sin(5*deg(x))};
\end{axis}
\end{tikzpicture}
```



Permettendo (spesso, ma non sempre) di diminuire il valore di `samples`, `smooth` può ridurre o evitare del tutto le probabilità di eccedere la memoria di calcolo del programma (che reagisce arrestando la composizione), ciò che troppi grafici ad altissima risoluzione in uno stesso documento potrebbero causare: la documentazione del pacchetto spiega come risolvere questi problemi.

Altri due esempi:

```
\begin{tikzpicture}
\begin{axis}
[axis lines=middle,
enlargelimits]
\addplot
[domain=-1:1,samples=200,smooth,
thick,blue]
{x*sin(deg(1/x))};
\end{axis}
\end{tikzpicture}
```

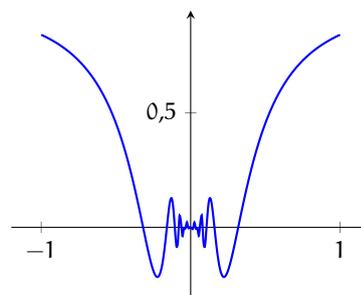


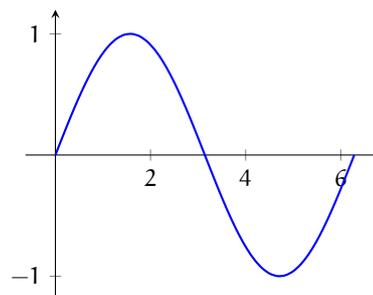
Tabella 55: Tratti disponibili in pgfplots.

Chiave	Risultato	Chiave	Risultato
solid	————	dashdotted	- . - . - .
dotted	densely dashdotted	- . - . - .
densely dotted	loosely dashdotted	- . - . - .
loosely dotted	dashdotdotted	- . - . - .
dashed	- - - - -	densely dashdotdotted	- . - . - .
densely dashed	- - - - -	loosely dashdotdotted	- . - . - .
loosely dashed	- - - - -		

```

\begin{tikzpicture}
\begin{axis}[axis lines=middle,
enlargelimits]
\addplot
[domain=0:2*pi,samples=40,smooth,
thick,blue]
{sin(deg(x))};
\end{axis}
\end{tikzpicture}

```



Si noti quanto segue.

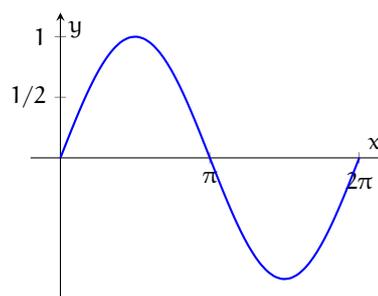
- La chiave `enlargelimits`, da mettere *dopo* `axis lines`, aumenta di poco la lunghezza di *entrambi* gli assi oltre i limiti stabiliti da `xmax` e `ymax` (`enlarge x limits` e `enlarge y limits` permettono di farlo separatamente), evitando che la curva oltrepassi i limiti del sistema di riferimento o finisca proprio sulla freccia di uno degli assi, oppure ancora solo per dare respiro al grafico.
- La chiave `domain`, per impostazione predefinita pari a $[-5, 5]$, imposta il dominio, nel primo esempio pari a $[-1, 1]$ e nel secondo a $[0, 2\pi]$.

Ora si ripropone l'esempio precedente con qualche variante:

```

\begin{tikzpicture}
\begin{axis}[axis lines=middle,
enlargelimits,
xtick={3.14,6.28},ytick={0.5,1},
xticklabels={\pi, 2\pi},
yticklabels={1/2, 1},
xlabel=$x$,ylabel=$y$]
\addplot [domain=0:2*pi,
samples=40,smooth,thick,blue]
{sin(deg(x))};
\end{axis}
\end{tikzpicture}

```

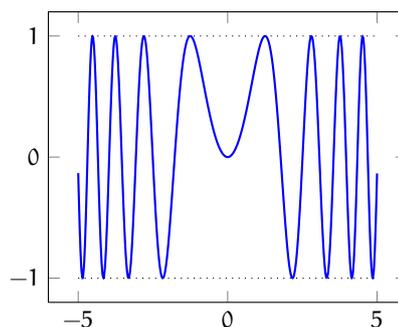


Si noti che:

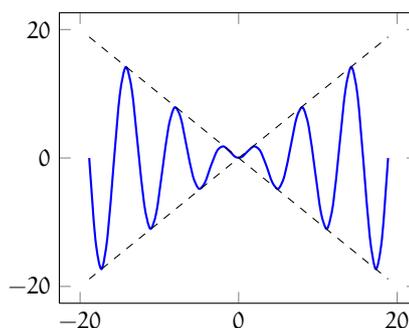
- con `xtick` e `ytick` si sceglie il valore delle tacche sugli assi x e y ;
- `xticklabels` e `yticklabels` producono le corrispondenti etichette se quelle automatiche non sono soddisfacenti.

Ecco un paio d'esempi con più `\addplot` consecutivi:

```
\begin{tikzpicture}
\begin{axis}
\addplot [samples=200,thick,
smooth,blue] {sin(deg(x^2))};
\addplot [dotted] {1};
\addplot [dotted] {-1};
\end{axis}
\end{tikzpicture}
```



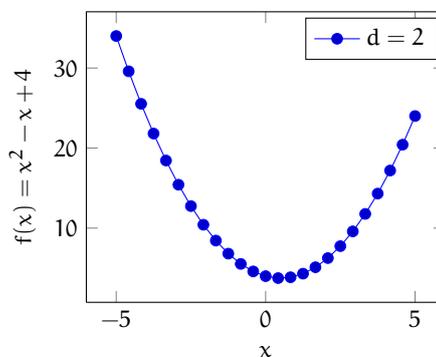
```
\begin{tikzpicture}
\begin{axis} [domain=-6*pi:6*pi]
\addplot [samples=50,smooth,
thick,blue] {x*sin(deg(x))};
\addplot [dashed] {x};
\addplot [dashed] {-x};
\end{axis}
\end{tikzpicture}
```



dove `dotted` disegna una linea punteggiata e `dashed` una linea tratteggiata (i tratti disponibili sono mostrati nella tabella 55 nella pagina precedente).

Il comando `\legend` produce una legenda:

```
\begin{tikzpicture}
\begin{axis}
[xlabel=$x$,
ylabel={$f(x)=x^2-x+4$}]
\addplot {x^2-x+4};
\legend{$d=2$}
\end{axis}
\end{tikzpicture}
```

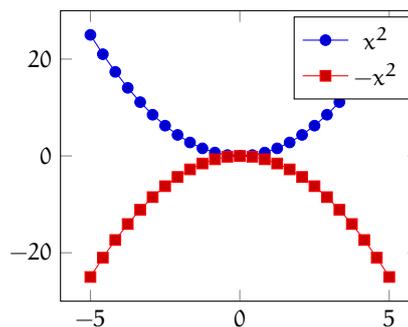


Si noti che:

- aspetto (riquadrata) e posizione (in alto a destra) della legenda sono predefiniti ma personalizzabili;
- se contiene caratteri particolari come = o la virgola, l'etichetta di un asse va racchiusa tra parentesi graffe;
- se è una formula matematica lunga, l'etichetta dell'asse y può rimanere nella posizione predefinita (in alternativa, si metta la formula nell'argomento di `title`).

Una legenda può contenere una voce per ciascun grafico tracciato:

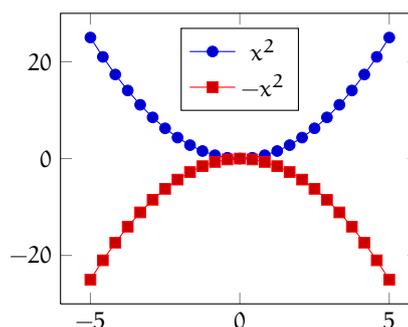
```
\begin{tikzpicture}
\begin{axis}
\addplot {x^2};
\addplot {-x^2};
\legend{$x^2$, $-x^2$}
\end{axis}
\end{tikzpicture}
```



Si noti che se in uno stesso sistema di riferimento ci sono più grafici, pgplots assegna automaticamente a ogni curva colore e marcatore distinti, scegliendoli secondo un ordine interno.

Come si vede, però, per impostazione predefinita la legenda compare sempre *dentro* il sistema di riferimento (se cartesiano) e può sovrapporsi al grafico nascondendone una parte. Si risolve il problema spostandola (la si può mettere in qualunque posizione dentro e fuori il sistema), come si mostra nell'esempio seguente:

```
\begin{tikzpicture}
\begin{axis}
[legend style={anchor=north,
at={(0.5,0.95)}}]
\addplot {x^2};
\addplot {-x^2};
\legend{$x^2$, $-x^2$}
\end{axis}
\end{tikzpicture}
```



Qualche prova e la lettura della documentazione del pacchetto permetteranno d'ottenere il risultato desiderato. Si noti che:

- `legend style` modifica lo stile predefinito della legenda;
- `anchor` specifica uno dei punti d'ancoraggio predefiniti per il riquadro della legenda (coincidenti essenzialmente con le direzioni d'una rosa dei venti a otto punte più un punto per il centro): qui `north` indica il punto medio del lato superiore;
- `at` definisce le coordinate del punto d'ancoraggio.

10.3.2 Curve in forma parametrica

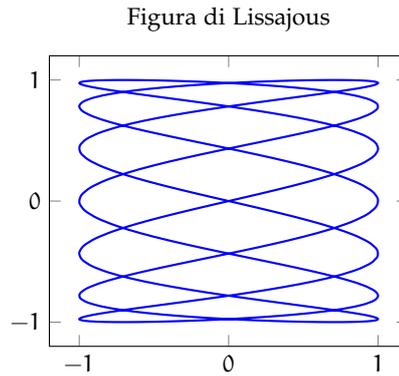
Curve nel piano

Di seguito si mostrano alcuni esempi di curve nel piano.

```

\begin{tikzpicture}
\begin{axis}
[title={Figura di Lissajous}]
\addplot
[domain=0:360,variable=\t,
samples=200,smooth,thick,blue]
({sin(7*t)},{sin(2*t)});
\end{axis}
\end{tikzpicture}

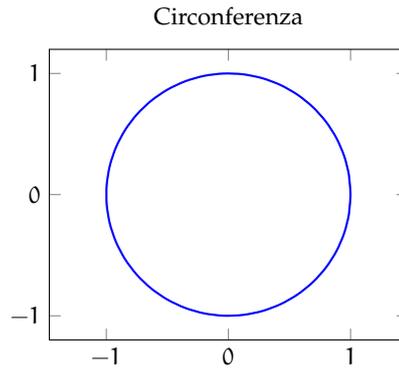
```



```

\begin{tikzpicture}
\begin{axis} [axis equal,
title={Circonferenza}]
\addplot
[domain=0:360,variable=\t,
samples=40,smooth,thick,blue]
({cos(t)},{sin(t)});
\end{axis}
\end{tikzpicture}

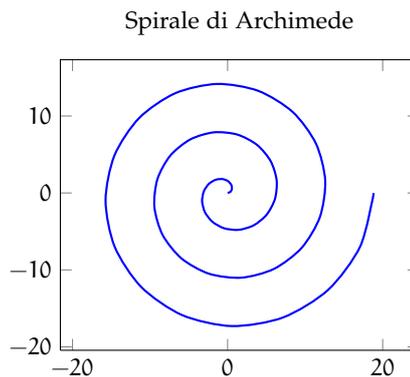
```



```

\begin{tikzpicture}
\begin{axis} [axis equal,
title={Spirale di Archimede}]
\addplot
[domain=0:6*pi,variable=\t,
samples=50,smooth,thick,blue]
({t*cos(deg(t))},
{t*sin(deg(t))});
\end{axis}
\end{tikzpicture}

```



Si noti che:

- `axis equal` imposta la stessa unità di misura su entrambi gli assi;
- le istruzioni per le curve parametriche nel piano vanno date nella forma

```

({x}, {y});

```

dove x e y sono funzioni del parametro;

- `variable` imposta il parametro (t , nei casi considerati), che nella propria definizione va preceduto da una barra rovescia.

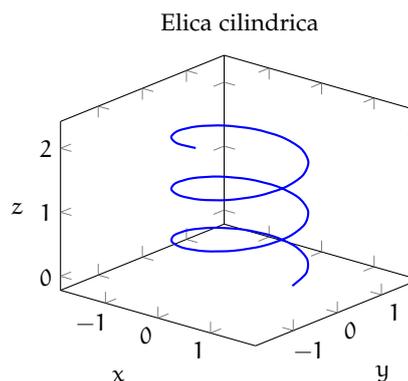
Curve nello spazio

Ecco un esempio di curva nello spazio tridimensionale:

```

\begin{tikzpicture}
\begin{axis}
[view={40}{20},axis equal,
xlabel=$x$,ylabel=$y$,
zlabel=$z$,
zlabel style={rotate=-90},
title={Elica cilindrica}]
\addplot3
[domain=0:5.5*pi,variable=\t,
samples=40,samples y=0,
smooth,thick,blue]
({cos(deg(t))},{sin(deg(t))},
{2*t/(5*pi)});
\end{axis}
\end{tikzpicture}

```



Si noti quanto segue.

- Il comando per tracciare grafici nello spazio è `\addplot3`, che richiede la sintassi spiegata nel paragrafo 10.2.2 a pagina 154.
- La chiave `view` imposta il punto di vista dell'osservatore (per una sua descrizione più completa si veda il paragrafo successivo).
- La chiave `zlabel` produce l'etichetta dell'asse z , ruotata con l'opzione descritta nel paragrafo 10.2.1 a pagina 150 per una maggiore leggibilità (in un grafico di questo tipo, in genere l'etichetta dell'asse y è già dritta).
- L'opzione `samples y=0` dichiara che si tratta di una curva e non di una superficie.
- Le istruzioni per le curve parametriche nello spazio richiedono la stessa sintassi già vista per quelle nel piano.

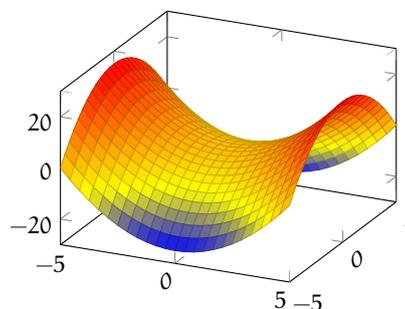
10.3.3 Funzioni reali di due variabili reali

Un paraboloide iperbolico per cominciare:

```

\begin{tikzpicture}
\begin{axis}
\addplot3 [surf]
{x^2-y^2};
\end{axis}
\end{tikzpicture}

```



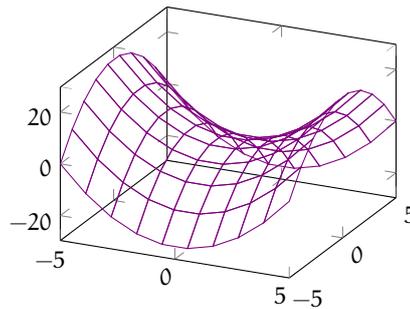
Si noti che `surf` disegna una superficie, visualizzata con colori predefiniti.

La chiave `mesh` produce grafici "a rete":

```

\begin{tikzpicture}
\begin{axis}
\addplot3
[mesh,samples=10,violet]
{x^2-y^2};
\end{axis}
\end{tikzpicture}

```

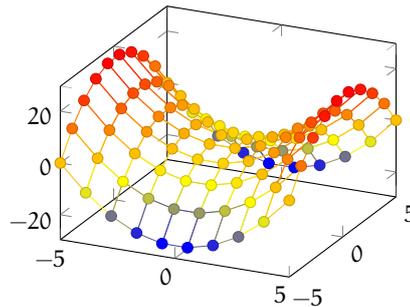


Aggiungendole scatter i nodi sono evidenziati con marcatori:

```

\begin{tikzpicture}
\begin{axis}
\addplot3
[mesh,scatter,samples=10]
{x^2-y^2};
\end{axis}
\end{tikzpicture}

```

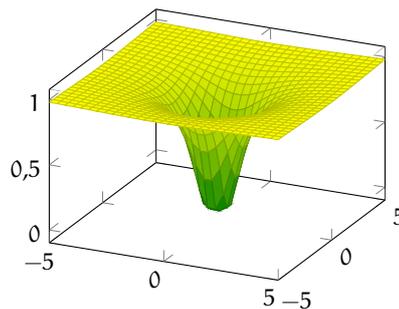


Ecco un altro esempio:

```

\begin{tikzpicture}
\begin{axis}
\addplot3 [samples=30,surf,
colormap/greenyellow]
{exp(-1/(x^2+y^2))};
\end{axis}
\end{tikzpicture}

```



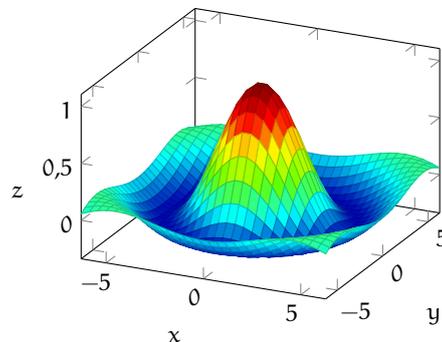
Si noti che `colormap/`*colorazione* personalizza i colori predefiniti (si veda la documentazione del pacchetto per le possibilità disponibili, alcune delle quali verranno mostrate nei prossimi esempi).

Ora due varianti di uno stesso grafico ottenute modificando i valori di `domain`, che imposta il dominio della funzione. Nel primo:

```

\begin{tikzpicture}
\begin{axis} [xlabel=$x$,
ylabel=$y$,zlabel=$z$,
zlabel style={rotate=-90}]
\addplot3 [domain=-2*pi:2*pi,
samples=30,surf,
colormap/bluered]
{sin(deg(sqrt(x^2+y^2)))/%
sqrt(x^2+y^2)};
\end{axis}
\end{tikzpicture}

```



il dominio della funzione è $[-2\pi, 2\pi] \times [-2\pi, 2\pi]$: specificando solo `domain`, con `domain=A` il dominio della funzione è il quadrato $A \times A$.

Nel secondo:

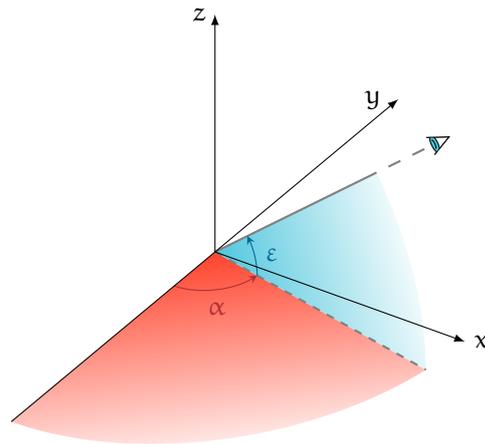
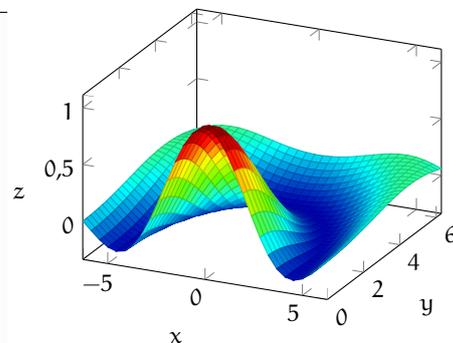


Figura 14: Definizione di azimut (α) ed elevazione (ϵ) secondo la convenzione adottata da pgfplots. Nel caso mostrato: $\alpha = 70^\circ$ e $\epsilon = 35^\circ$.

```

\begin{tikzpicture}
\begin{axis} [xlabel=$x$,
ylabel=$y$,zlabel=$z$,
zlabel style={rotate=-90}]
\addplot3 [domain=-2*pi:2*pi,
y domain=0:2*pi,samples=30,
surf,colormap/bluered]
{sin(deg(sqrt(x^2+y^2)))/%
sqrt(x^2+y^2)};
\end{axis}
\end{tikzpicture}

```



il dominio della funzione è $[-2\pi, 2\pi] \times [0, 2\pi]$: specificando anche `y domain`, con `domain=A` e `y domain=B` il dominio della funzione è il rettangolo $A \times B$.

Il punto di vista: la chiave view

La chiave `view` imposta il punto di vista dell'osservatore e richiede la seguente sintassi generale:

```
view={⟨azimut⟩}{⟨elevazione⟩}
```

dove i due valori richiesti (pari a 25 e 30 per impostazione predefinita) indicano rispettivamente azimut ed elevazione nel sistema di coordinate sferiche. La figura 14 mostra come pgfplots gestisce il punto di vista.

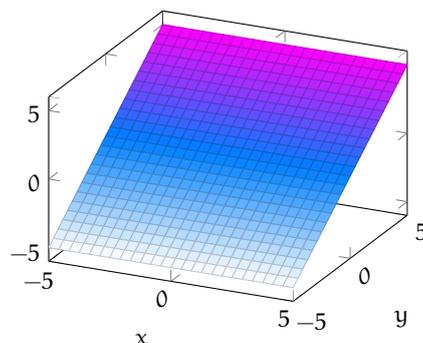
Seguono ora alcune varianti notevoli di uno stesso piano ottenute modificando i valori di `view`.

```

\begin{tikzpicture}
\begin{axis} [view={25}{30},
title={Punto di vista
predefinito},xlabel=$x$,
ylabel=$y$,zlabel=$z$,
zlabel style={rotate=-90}]
\addplot3 [surf,colormap/cool]
{y};
\end{axis}
\end{tikzpicture}

```

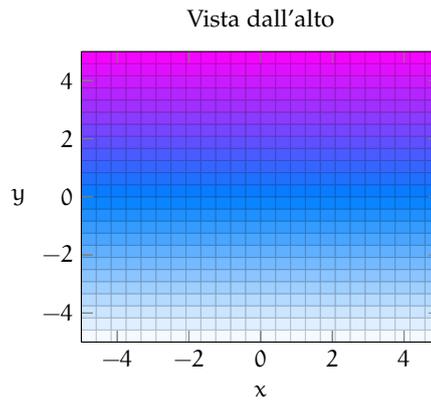
Punto di vista predefinito



```

\begin{tikzpicture}
\begin{axis} [view={0}{90},title=
{Vista dall'alto},xlabel=$x$,
ylabel=$y$,zlabel=$z$,
ylabel style={rotate=-90}]
\addplot3 [surf,colormap/cool]
{y};
\end{axis}
\end{tikzpicture}

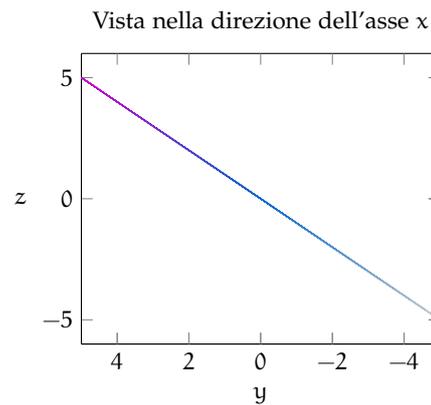
```



```

\begin{tikzpicture}
\begin{axis} [view={270}{0},
title={Vista nella direzione
dell'asse $x$},xlabel=$x$,
ylabel=$y$,zlabel=$z$,
zlabel style={rotate=-90}]
\addplot3 [surf,colormap/cool]
{y};
\end{axis}
\end{tikzpicture}

```



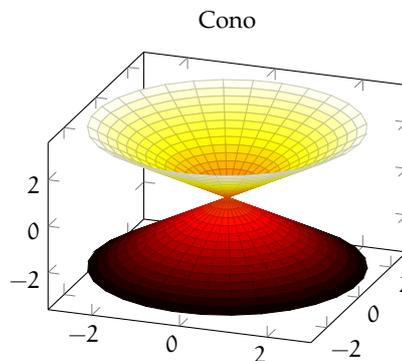
10.3.4 Superfici in forma parametrica

Un esempio:

```

\begin{tikzpicture}
\begin{axis}
[view={20}{30},title={Cono}]
\addplot3
[domain=-3:3,y domain=0:360,
variable=\u,variable y=\v,
samples=30,z buffer=sort,
surf,colormap/hot2]
({u*cos(v)}, {u*sin(v)}, u);
\end{axis}
\end{tikzpicture}

```



Si noti che:

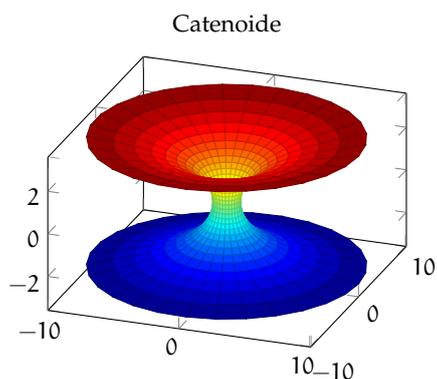
- variable y imposta il secondo parametro coordinato (v , in questo caso);
- la chiave `z buffer` suggerisce a `pgfplots` i criteri da seguire per proiettare i punti dello spazio tridimensionale sul quadro di proiezione (in questo caso, `sort` traccia per primi i segmenti più distanti dal punto di osservazione).

Di seguito si mostra una galleria d'esempi.

```

\begin{tikzpicture}
\begin{axis}
[view={20}{35},title={Catenoide}]
\addplot3
[domain=0:360,y domain=-3:3,
variable=\u,variable y=\v,
samples=30,z buffer=sort,
surf,colormap/jet]
({cos(u)*cosh(v)},
{sin(u)*cosh(v)},v);
\end{axis}
\end{tikzpicture}

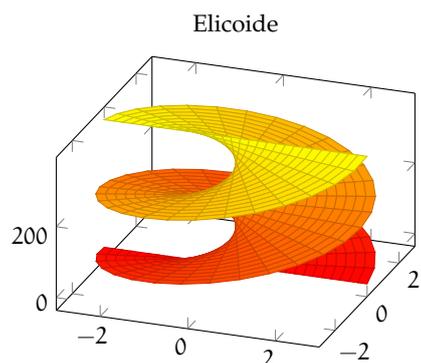
```



```

\begin{tikzpicture}
\begin{axis}
[view={20}{35},title={Elicoide}]
\addplot3
[domain=-3:3,y domain=0:360,
variable=\u,variable y=\v,
samples=30,z buffer=sort,
surf,colormap/redyellow]
({u*cos(v)}, {u*sin(v)},v);
\end{axis}
\end{tikzpicture}

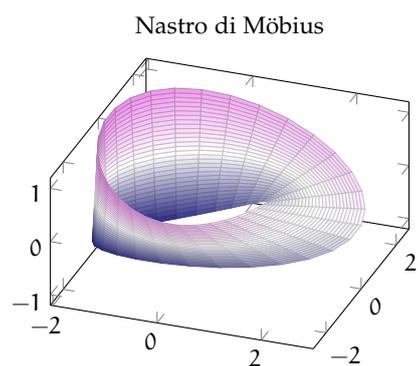
```



```

\begin{tikzpicture}
\begin{axis}
[view={20}{45},
title={Nastro di M\"obius}]
\addplot3
[domain=0:360,y domain=-1:1,
variable=\u,variable y=\v,
samples=30,z buffer=sort,
surf,colormap/violet]
({2*cos(u)+v*cos(u)*cos(u/2)},
{2*sin(u)+v*sin(u)*sin(u/2)},
{v*sin(u/2)});
\end{axis}
\end{tikzpicture}

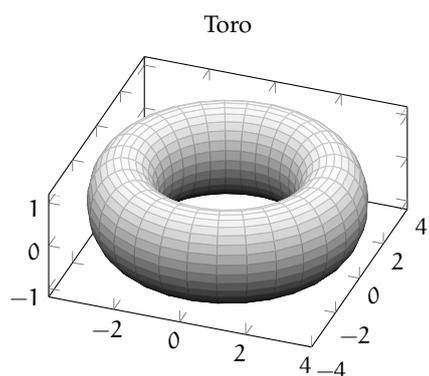
```



```

\begin{tikzpicture}
\begin{axis}
[view={20}{55},title={Toro}]
\addplot3
[domain=0:360,y domain=0:360,
variable=\u,variable y=\v,
samples=30,z buffer=sort,
surf,colormap/blackwhite]
({(3+cos(u))*cos(v)},
{(3+cos(u))*sin(v)},
{sin(u)});
\end{axis}
\end{tikzpicture}

```



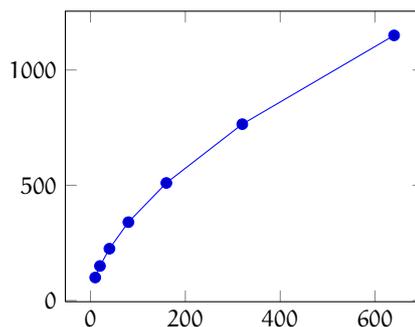
10.4 CURVE E SUPERFICI DATE PER COORDINATE

I prossimi sono due esempi di curve date per coordinate:

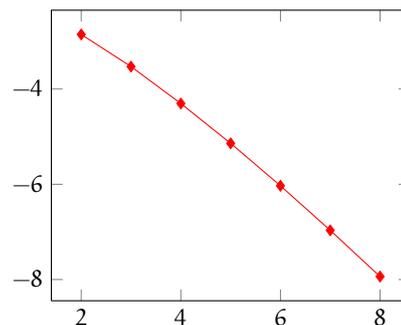
Tabella 56: Alcuni marcatori disponibili in pgplots.

Chiave	Risultato	Chiave	Risultato
*		square	
o		square*	
x		halfsquare*	
+		triangle	
asterisk		triangle*	
star		diamond	
oplus		diamond*	
otimes		halfdiamond*	

```
\begin{tikzpicture}
\begin{axis}
\addplot coordinates
{(10, 100) (20, 150)
(40, 225) (80, 340)
(160, 510) (320, 765)
(640, 1150)};
\end{axis}
\end{tikzpicture}
```



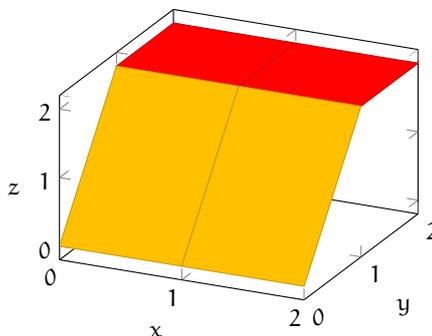
```
\begin{tikzpicture}
\begin{axis}
\addplot [red,mark=diamond*]
coordinates
{(2, -2.855) (3, -3.530)
(4, -4.305) (5, -5.141)
(6, -6.032) (7, -6.967)
(8, -7.937)};
\end{axis}
\end{tikzpicture}
```



Si noti che `mark` imposta il tipo di marcatore, nel secondo esempio un rombo. In alternativa, se ne può scegliere un altro tra quelli mostrati nella tabella 56 o definirne uno personale.

Infine un esempio di superficie data per coordinate:

```
\begin{tikzpicture}
\begin{axis} [xlabel=$x$,
ylabel=$y$,zlabel=$z$,
zlabel style={rotate=-90}]
\addplot3 [surf] coordinates
{(0,0,0) (1,0,0) (2,0,0)
(0,1,2) (1,1,2) (2,1,2)
(0,2,2) (1,2,2) (2,2,2)};
\end{axis}
\end{tikzpicture}
```



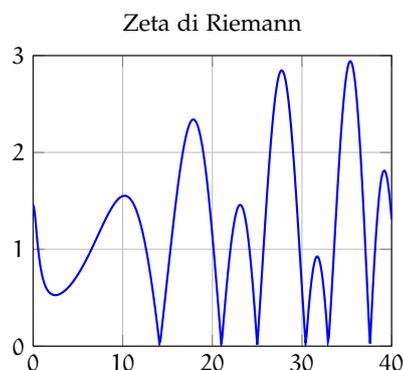
Si noti che in questo caso la sequenza delle coordinate assume la struttura di una matrice, nella quale ogni riga di valori va separata da quella successiva con una riga vuota.

10.5 CURVE E SUPERFICI CAMPIONATE DA FILE

A titolo d'esempio, si mostrano qui tre grafici (i primi due nel piano, il terzo nello spazio) costruiti con l'aiuto di file esterni ottenuti nei modi spiegati nel paragrafo 10.2.2 a pagina 154.

Il primo esempio mostra il grafico del valore assoluto della funzione Zeta di Riemann sulla retta $\text{Re } z = 1/2$.

```
\begin{tikzpicture}
\begin{axis} [xmin=0,xmax=40,
ymin=0,ymax=3,grid=major,
title={Zeta di Riemann}]
\addplot [thick,blue]
file {zeta.txt};
\end{axis}
\end{tikzpicture}
```

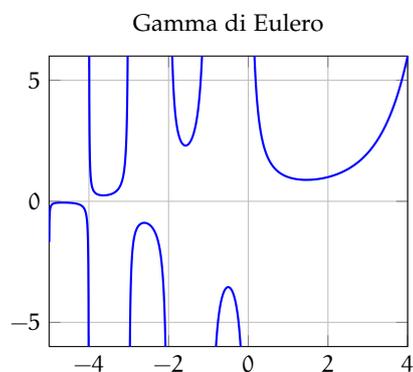


Di seguito si mostrano le prime righe del file `zeta.txt`:

```
0.0  1.460354508809587
0.1  1.433807867750897
0.2  1.362770945580488
0.3  1.266515016158303
```

Il prossimo esempio mostra il grafico della funzione Gamma di Eulero:

```
\begin{tikzpicture}
\begin{axis} [xmin=-5,xmax=4,
ymin=-6,ymax=6,grid=major,
title={Gamma di Eulero}]
\addplot
[unbounded coords=jump,
thick,blue]
file {gamma.txt};
\end{axis}
\end{tikzpicture}
```



dove l'opzione `unbounded coords=jump` gestisce i punti di discontinuità. Ed ecco le prime righe del file `gamma.txt`:

```
-5.000 NaN
-4.995 -1.6810104460206580
-4.990 -0.8478047198471037
-4.985 -0.5701560523263895
```

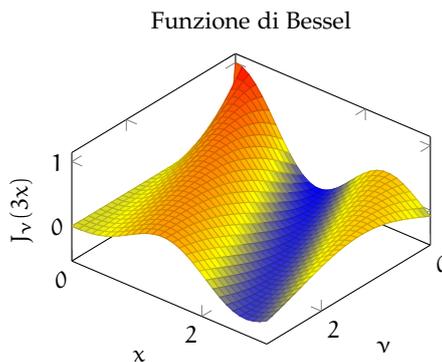
Si noti che le righe contenenti i valori NaN (*Not a Number*) e inf (*infinity*), indicanti rispettivamente un valore non numerico e infinito, sono sempre ignorate da `pgfplots`.

Il prossimo grafico, infine, mostra la funzione di Bessel (di prima specie):

```

\begin{tikzpicture}
\begin{axis} [view={130}{50},
xlabel=$\nu$,ylabel=$x$,
zlabel={$J_\nu(3x)$},
title={Funzione di Bessel}]
\addplot3
[surf,z buffer=sort]
file {bessel.txt};
\end{axis}
\end{tikzpicture}

```



Di seguito si mostrano le prime righe del file `bessel.txt`:

```

0.000000 0.000000 1.000000
0.000000 0.103448 0.976066
0.000000 0.206897 0.905981
0.000000 0.310345 0.794755

```

Si noti che in questo caso le terne di coordinate che rappresentano i punti devono rispettare un ordine ben preciso ed essere in un formato opportuno (la documentazione del pacchetto spiega come farlo).

10.6 ALTRI SISTEMI DI RIFERIMENTO

In questa sezione si mostrano esempi degli altri sistemi di riferimento definiti da `pgplots`. Alcuni di essi richiedono di caricare *nel preambolo* la libreria indicata con:

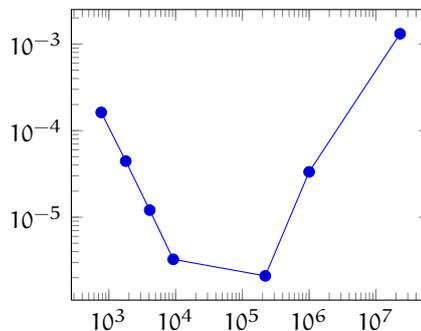
```
\usepgfplotslibrary{libreria}
```

Per cominciare, un esempio di piano cartesiano logaritmico:

```

\begin{tikzpicture}
\begin{loglogaxis}
\addplot coordinates
{(769, 1.6227e-04)
(1793, 4.4425e-05)
(4097, 1.2071e-05)
(9217, 3.2610e-06)
(2.2e5, 2.1E-6)
(1e6, 0.00003341)
(2.3e7, 0.00131415)};
\end{loglogaxis}
\end{tikzpicture}

```

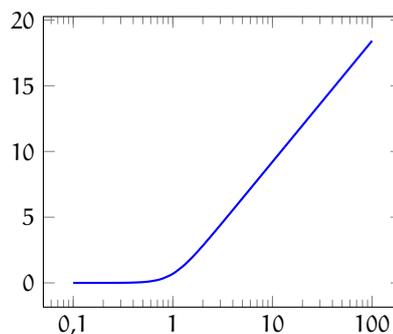


Ora un grafico con ascissa logaritmica:

```

\begin{tikzpicture}
\begin{semilogxaxis}
[log ticks with fixed point]
\addplot
[domain=0.1:100,
thick,blue,smooth]
{\ln(1+x^4)};
\end{semilogxaxis}
\end{tikzpicture}

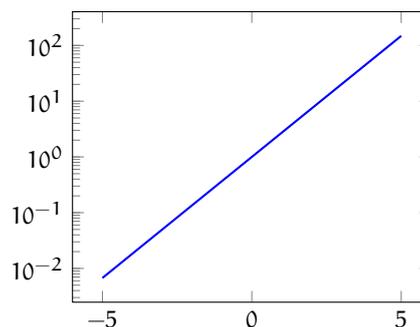
```



L'opzione `log ticks with fixed point` evita marcatori di tacca con esponenti.

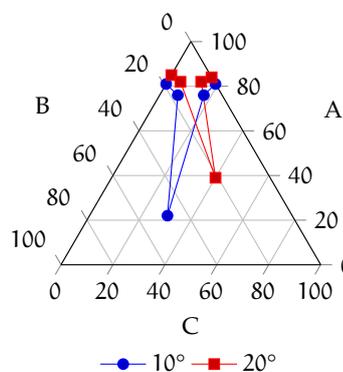
Il prossimo è un grafico con ordinata logaritmica:

```
\begin{tikzpicture}
\begin{semilogyaxis}
\addplot [thick,blue]
{exp(x)};
\end{semilogyaxis}
\end{tikzpicture}
```



Segue un esempio di diagramma ternario (richiede la libreria `ternary`):

```
\begin{tikzpicture}
\begin{ternaryaxis}
[xlabel=A,ylabel=B,zlabel=C,
legend style={anchor=north,
at={(0.5,-0.35)},draw=none},
legend columns=-1]
\addplot3 coordinates
{(0.81, 0.19, 0.00)
(0.76, 0.17, 0.07)
(0.22, 0.40, 0.30)
(0.76, 0.07, 0.17)
(0.81, 0.00, 0.19)};
\addplot3 coordinates
{(0.85, 0.15, 0.00)
(0.82, 0.13, 0.05)
(0.39, 0.30, 0.40)
(0.82, 0.06, 0.13)
(0.84, 0.00, 0.16)};
\legend{$10^\text{degree}$,
$20^\text{degree}$}
\end{ternaryaxis}
\end{tikzpicture}
```

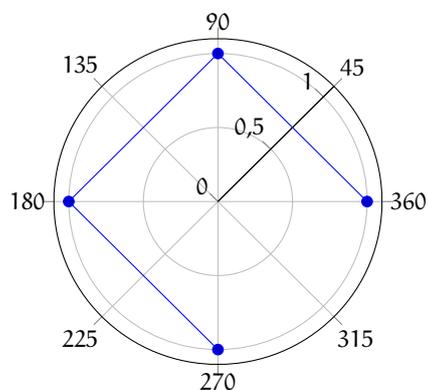


Si noti che;

- `draw=none` elimina il riquadro della legenda;
- `legend columns=-1` ne dispone gli elementi orizzontalmente.

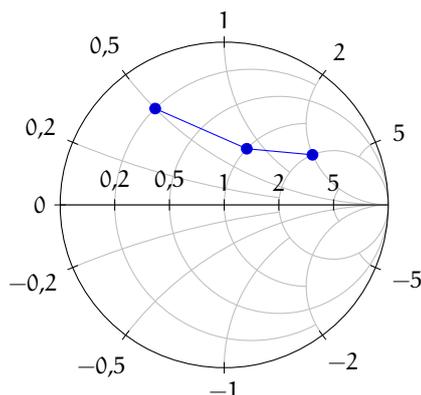
Il prossimo è un sistema di coordinate polari (richiede la libreria `polar`):

```
\begin{tikzpicture}
\begin{polaraxis}
[xmin=45,xmax=405]
\addplot coordinates
{(0, 1) (90, 1)
(180, 1) (270, 1)};
\end{polaraxis}
\end{tikzpicture}
```



Infine una carta di Smith (richiede la libreria smithchart):

```
\begin{tikzpicture}
\begin{smithchart}
\addplot coordinates
{(0.2, 0.5) (1, 0.8) (2, 2)};
\end{smithchart}
\end{tikzpicture}
```



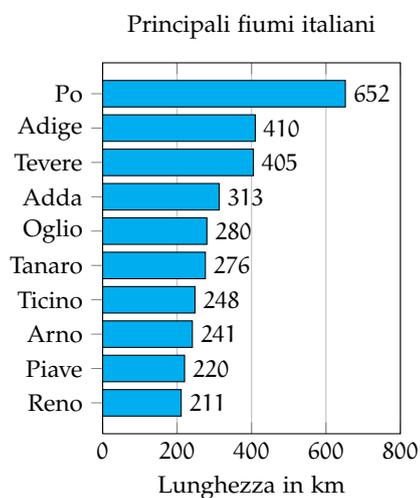
10.7 DIAGRAMMI A BARRE

Con pgfplots si possono realizzare anche diagrammi a barre. L'*ortogramma* si distingue dall'*istogramma* non solo per il diverso orientamento delle barre, ma anche perché di solito queste sono staccate tra di loro.

10.7.1 Ortogrammi

Un ortogramma:

```
\begin{tikzpicture}
\begin{axis}
[xbar,xmin=0,xmax=800,
height=6.5cm,
xmajorgrids=true,
ytick pos=left,
title={Principali fiumi
italiani},
xlabel={Lunghezza in km},
symbolic y coords={Reno,Piave,
Arno,Ticino,Tanaro,Oglio,Adda,
Tevere,Adige,Po},
ytick=data,nodes near coords,
nodes near coords align=%
{horizontal}]
\addplot
[fill=cyan,draw=black]
coordinates
{(211,Reno) (220,Piave)
(241,Arno) (248,Ticino)
(276,Tanaro) (280,Oglio)
(313,Adda) (405,Tevere)
(410,Adige) (652,Po)};
\end{axis}
\end{tikzpicture}
```



Si noti che:

- xbar produce le barre orizzontali;
- xmajorgrids produce una "griglia" di sole linee verticali;
- ytick pos imposta la posizione delle tacche sull'asse y;

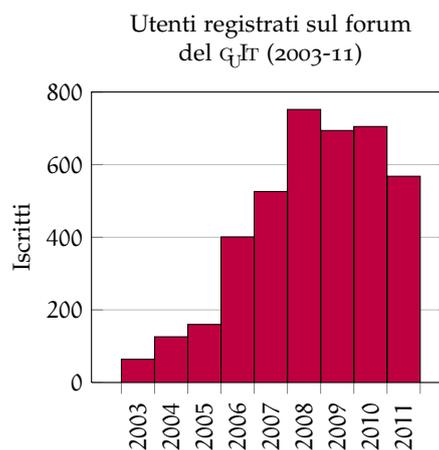
- `symbolic y coords` produce il proprio argomento come etichetta della barra;
- `data` attribuisce alle etichette i valori dichiarati come ordinate;
- `node near coords` mette il valore indicato nelle coordinate alla fine di ogni barra;
- `node near coords align` ne aggiusta l'allineamento in modo che non si sovrapponga alla barra;
- `fill` imposta il colore che riempirà le barre;
- `draw` ne imposta il colore di contorno.

Si possono creare ortogrammi anche con il pacchetto `bchart` (se ne veda la documentazione), che richiede una sintassi più semplice.

10.7.2 Istogrammi

Un istogramma:

```
\begin{tikzpicture}
\begin{axis} [ylabel={Iscritti},
ybar,ymin=0,ymax=800,xtick=data,
ymajorgrids=true,xtick pos=left,
x tick label style={
{rotate=90,anchor=east},
xticklabel interval boundaries,
symbolic x coords={$2003$,
$2004$, $2005$, $2006$, $2007$,
$2008$, $2009$, $2010$, $2011$,
$2012$},
title style={align=center},
title={Utenti registrati sul
forum\ del \GuIT{ } (2003-11)}]
\addplot
[ybar interval,fill=purple,
draw=black] coordinates
{($2003$, 64) ($2004$, 126)
($2005$, 160) ($2006$, 401)
($2007$, 526) ($2008$, 752)
($2009$, 694) ($2010$, 705)
($2011$, 568) ($2012$, 0)};
\end{axis}
\end{tikzpicture}
```



Alle osservazioni sul grafico precedente, qui riferibili all'asse delle y , si aggiungano le seguenti:

- `xtick pos` imposta la posizione delle tacche sull'asse x ;
- `x tick label style` imposta l'aspetto delle etichette delle colonne (qui ruotate di 90° in senso antiorario);
- `xticklabel interval boundaries` centra l'etichetta di ogni colonna;
- `title style` imposta lo stile del titolo, necessario per averlo su due righe;
- `ybar interval` rende adiacenti le colonne, altrimenti staccate;
- un istogramma richiede di definire una colonna *in più* rispetto a quelle necessarie (qui è quella relativa all'anno 2012).

Prima o poi, capita a tutti di aver bisogno di “cose” non contemplate da \LaTeX standard: di solito si tratta di comandi e ambienti ad hoc o personalizzazioni di quelli esistenti. In questo capitolo s’impara come farlo.

11.1 COMANDI E AMBIENTI PERSONALI

11.1.1 Nuovi comandi

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

\LaTeX risolve questi problemi molto più “filosoficamente” di quanto non facciano altri programmi. Anziché agire a mano, si può definire *una volta per tutte* un nuovo comando, che si chiamerà `\piana`, che produce il proprio argomento (un nome di pianta, in questo caso) nello stile deciso dall’utente (in corsivo, in questo caso). Se l’imposizione del nero avviene a documento completato, basterà modificare *una volta per tutte* la definizione del comando e non più tutti i `\piana` nel documento.

Il nuovo comando, dunque, non descrive come l’argomento debba essere reso, ma come lo si è pensato. Perciò in un documento scritto con \LaTeX i comandi per cambiare lo stile vanno usati molto di rado, in favore di comandi che rispecchino la logica di ciò che si sta scrivendo.

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

```
\newcommand{⟨nome⟩}[⟨numero di argomenti⟩]{⟨definizione⟩}
```

Dove:

- `⟨nome⟩` è il nome che si dà al nuovo comando.
- `⟨numero di argomenti⟩` è il numero di argomenti obbligatori che gli si assegna, fino a un massimo di nove. Non specificando questo valore, il nuovo comando non avrà argomenti.
- `⟨definizione⟩` sono le istruzioni che specificano ciò che si vuole che il nuovo comando “faccia”. Gli argomenti eventualmente assegnati al comando s’indicano nella `⟨definizione⟩` con `#` seguito da un numero progressivo: `#1`, `#2`, eccetera.

Il comando appena esaminato serve a definire nuovi comandi senza e con argomenti. L’esempio seguente mostra la sintassi di un comando personale *senza argomenti*:

```
\newcommand{\arte}{\emph{L’arte di scrivere con \LaTeX}}
```

utile, per esempio, se in un documento si dovesse scrivere ripetutamente il titolo di questa guida (si noti che un comando di questo tipo “produce testo”, e va perciò terminato nei modi già visti nel paragrafo 3.4.1 a pagina 22):

Questo lavoro s'intitola <code>\arte</code> .	Questo lavoro s'intitola <i>L'arte di scrivere con L^AT_EX</i> .
---	--

Si faccia attenzione, però, a non esagerare con i comandi personali semplicemente per abbreviare il testo, come `\gb` per *Gran Bretagna*, per esempio: alla lunga rendono il codice illeggibile.

Di seguito si mostra la sintassi di un comando personale *con argomenti*:

<code>\newcommand{\pianta}[1]{\textit{#1}}</code>

Si è assegnato al nuovo comando un solo argomento obbligatorio ([1]). Quando si dà il comando, accade questo: il testo nell'argomento di `\pianta` viene “passato” a #1 e trattato secondo la *definizione*. In questo caso verrà reso in corsivo, come si vede nell'esempio seguente:

<code>\pianta{Rosa canina}</code>	<i>Rosa canina</i>
-----------------------------------	--------------------

In modo analogo, anche se non è molto frequente, si possono definire nuovi comandi con più argomenti (specificando [2], [3], eccetera).

L^AT_EX impedisce di definire un comando già esistente, ma non di ridefinirlo: per farlo, si usa `\renewcommand`, che ha la stessa sintassi di `\newcommand`.

A chi volesse approfondire gli aspetti della programmazione avanzata di L^AT_EX si consiglia la lettura di [Gregorio, 2009].

11.1.2 Nuovi ambienti

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

<code>\newenvironment{<nome>}[<numero di argomenti>]% {<comandi di apertura>}{<comandi di chiusura>}</code>

Dove:

- *<nome>* è il nome che si dà al nuovo ambiente.
- *<numero di argomenti>* è il numero di argomenti che gli si assegna. Non specificando questo valore, il nuovo ambiente non avrà argomenti.
- *<comandi di apertura>* sono le istruzioni, che usano gli argomenti eventualmente presenti, da eseguire all'inizio dell'ambiente.
- *<comandi di chiusura>* sono le istruzioni da eseguire alla chiusura dell'ambiente.

Si noti che si possono assegnare eventuali argomenti solo ai *<comandi di apertura>*. In pratica, definire un nuovo ambiente equivale a definire due nuovi comandi, uno di apertura e uno di chiusura, da usare come al solito.

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

<code>\newenvironment{itaitemize}{\begin{itemize}\itshape}{\end{itemize}}</code>
--

si potrà usare il nuovo ambiente `itaitemize` così:

<code>\begin{itaitemize}</code>	
<code>\item Un elenco con voci\dots</code>	• <i>Un elenco con voci...</i>
<code>\item \dots</code> automaticamente in corsivo.	• <i>... automaticamente in corsivo.</i>
<code>\end{itaitemize}</code>	

Per ridefinire ambienti già esistenti, si usa `\renewenvironment` (analogo a `\renewcommand`), che ha la stessa sintassi di `\newenvironment`.

Chi volesse cimentarsi nella costruzione di interi pacchetti o classi personali, trova le istruzioni nel file `clsguide.pdf`, recuperabile come al solito.

11.2 PERSONALIZZARE IL TESTO

11.2.1 Font

In tipografia la parola *font* indica un insieme eterogeneo di caratteri (detti anche *glifi*) accomunati da un certo stile grafico. Più font con caratteristiche comuni costituiscono una *famiglia di font*, che comprende senz'altro la variante regolare (detta anche *tondo*) e almeno il *corsivo*, il **MAIUSCOLETTO** e il **nero**. Tra le caratteristiche più importanti di un font ci sono la *larghezza* e la presenza o meno delle *grazie*.

I caratteri dei font a larghezza fissa (o *monospaced* o *typewriter*, usati nei codici d'esempio di questa guida) hanno tutti la stessa larghezza e sono adatti per scrivere in un editor di testi o per incolonnare dati numerici, caratteristica che non hanno i font a larghezza *variabile*, più versatili e indicati per riempire la riga in modo ottimale.

I caratteri di un font *con grazie* (o *serif*, usati di solito nel testo principale) presentano piccole rifiniture alle estremità che alla lunga li rendono più leggibili su carta, al contrario di quelli di un font senza grazie (o *sans serif*, in questa guida usati per i nomi dei pacchetti), più indicati per la lettura a schermo o per corpi molto piccoli. Le figure 15 e 16 nella pagina seguente mostrano le quattro categorie di font appena esaminate.

Di regola, \LaTeX sceglie la variante appropriata in base alla struttura logica del documento (capitoli, paragrafi, testatine, eccetera), ma può accadere di doverne modificare a mano stile e corpo.

Scegliere i font

A volte, per i motivi più vari, potrebbe essere necessario usare font diversi da quelli predefiniti: per una precisa richiesta dell'editore, per aumentare la leggibilità del documento, per "alleggerire" il proprio lavoro di qualche pagina caricando un font più compatto, eccetera.

\LaTeX permette di usare praticamente *ogni tipo* di font in circolazione ma, si noti bene, un font *non* va usato per differenziare parti di testo (si usino per questo i comandi descritti nel paragrafo 4.4 a pagina 54) né tanto meno per abbellirle. Analogamente, non si considerino i vari stili come una "tavolozza" da cui attingere a piacere: tondo, corsivo e maiuscoletto, da usare secondo i criteri spiegati nell'appendice A a pagina 191, dovrebbero bastare per la gran parte delle esigenze.

In linea generale, in un documento servono quattro famiglie di font:

- una famiglia con grazie per il testo principale;
- una famiglia senza grazie per scopi particolari;
- una famiglia a larghezza fissa per indirizzi Internet, codici e alcune parole di ambito informatico;
- una famiglia per scrivere la matematica.

Scegliere famiglie che stiano bene insieme richiede abilità ed esperienza: perciò, *specie se si è alle prime armi*, si consiglia di ricorrere senz'altro a classi



Figura 15: Famiglie di font a larghezza variabile.



Figura 16: Famiglie di font a larghezza fissa.

e pacchetti (anche non standard, come ClassicThesis, ArsClassica o suftesi) che abbiano già risolto questo problema.

Si può vedere una panoramica (quasi) completa dei font latini liberamente disponibili per l'uso con L^AT_EX() già presenti in T_EX Live (completi delle istruzioni per caricarli) su [FONTS](#).

11.2.2 Lingue straniere

Se in un documento le parti in lingua straniera sono molte, anziché usare i comandi standard per le lingue descritti nel paragrafo 3.2.1 a pagina 19 è più conveniente ridefinirli come si spiega di seguito.

Si può ottenere una versione personale di `\foreignlanguage` con

```
\newcommand{\inglese}[1]{\foreignlanguage{english}{\em #1}}
```

da usare come mostra l'esempio seguente:

```
Come scrisse una volta Donald Knuth, <<\inglese{we have seen that computer programming is an art}>>.
```

```
Come scrisse una volta Donald Knuth, «we have seen that computer programming is an art».
```

Con le opportune sostituzioni si possono definire comandi per tutte le lingue di cui si ha bisogno.

In modo analogo si può ridefinire l'ambiente `otherlanguage*`

```
\newenvironment{inglese}%
{\begin{otherlanguage*}{english}\em}%
{\end{otherlanguage*}}
```

(o `otherlanguage`, se serve) da usare come al solito.

Anche l'ambiente `quoting` presentato nel paragrafo 4.8.3 a pagina 61 potrà essere ridefinito per eventuali citazioni in lingua straniera. Si noti che in tutti questi esempi si è usata la dichiarazione `\em` per mettere in evidenza il testo straniero, secondo le consuetudini italiane.

11.3 SPECIALITÀ

Il tutto è maggiore della somma delle parti.

Aristotele
Metafisica

In questo paragrafo si descrivono soltanto alcune delle numerose possibilità offerte da L^AT_EX per personalizzare il proprio documento. Per ulteriori modifiche, si rimanda al paragrafo 11.5 a pagina 182.

11.3.1 Epigrafi

Esigenze *molto* particolari potrebbero richiedere di scrivere un'epigrafe in testa a un capitolo o un paragrafo: per realizzarla facilmente c'è il pacchetto `epigraph`. Un'epigrafe come quella che si vede dopo il titolo di questa sezione si ottiene con il seguente codice:

```
\section{Specialità}
\epigraph{Il tutto è maggiore della somma delle parti}%
{Aristotele\ \emph{Metafisica}}
```

11.3.2 Capilettera

LE CONVENZIONI tipografiche permettono, a scopo decorativo, di sostituire la prima lettera di un capitolo con un *capolettera*, cioè una lettera di corpo maggiore delle altre (come qui). I capilettera hanno le forme più svariate: vanno dai caratteri nel font in uso a quelli elaboratissimi disegnati appositamente per pubblicazioni particolarmente ricercate.

Ne offre un vasto assortimento il pacchetto `lettrine` (se ne veda la documentazione), che per funzionare al meglio richiede a propria volta di caricare *prima* di `fontenc` con l'opzione `T1` il pacchetto `type1ec` (per scalare "a piacimento" i font standard di \LaTeX disegnati in vari corpi; usando font disegnati in un solo corpo come per esempio quelli dei pacchetti `txfonts`, `pxfonts`, `mathpazo` e `fourier`, invece, `type1ec` non serve):

```
\usepackage{type1ec}
\usepackage[T1]{fontenc}
\usepackage{lettrine}
```

Il comando `\lettrine`, la cui sintassi dovrebbe essere chiara, produce il capolettera:

```
\lettrine{<capolettera>}{<eventuale testo in maiuscoletto>}
```

Di seguito si riporta un esempio di testo con capolettera.

```
\lettrine{L}{e convenzioni}
tipografiche permettono, a scopo
decorativo, di sostituire la
prima lettera di un capitolo con
un \emph{capolettera}, cioè una
lettera di corpo maggiore delle
altre (come qui).
```

LE CONVENZIONI tipografiche permettono, a scopo decorativo, di sostituire la prima lettera di un capitolo con un *capolettera*, cioè una lettera di corpo maggiore delle altre (come qui).

Una consuetudine molto seguita prevede di sfumare il passaggio dal capolettera al testo normale con qualche parola in maiuscoletto, come si è fatto in questo paragrafo: si può fare, ma *non* è una regola.

11.3.3 Scritture curiose

Con \LaTeX , il pacchetto `shapepar` permette di "sagomare" un testo in molte forme diverse: cuori (come qui), quadrati, diamanti, cerchi, stelle a cinque punte, rettangoli, esagoni, candele che bruciano, triangoli diversamente orientati (si può anche ottenere un testo nella forma del logo \TeX). Basta scrivere il testo da sagomare nell'argomento del comando corrispondente (`\heartpar` in questo caso, se ne veda la documentazione). Forse questa possibilità non verrà molto usata, ma è un chiaro esempio di come con pochissima fatica si possano ottenere risultati impensabili con altri programmi. Avete notato il piccolo cuore proprio qui sotto?



11.3.4 Documenti multicolonna

Scrivere su più colonne

L'opzione di classe `twocolumn` permette di comporre l'intero documento su due colonne, ma ha il difetto di non bilanciare il testo nell'ultima pagina.

Il pacchetto `multicol` (se ne veda la documentazione) risolve il problema. Scrivendo semplicemente

```
\begin{multicols}{\langle numero \rangle}
\langle testo multicolonna \rangle
\end{multicols}
```

dove:

- $\langle numero \rangle$ è il numero delle colonne (da 2 a 10) in cui s'intende dividere il testo;

• $\langle testo multicolonna \rangle$ si spiega da sé; si ottiene un testo su più colonne, bilanciate automaticamente e sezionabili con i comandi consueti. Si noti che l'ambiente `multicols` può essere usato per un intero documento o solo per alcune sezioni di esso, e che non accetta note a margine né oggetti mobili che non siano a tutta pagina, ciò che lo rende adatto piuttosto a documenti di solo testo o a sezioni particolari come l'indice analitico.

Inserire oggetti mobili

L'inserimento di oggetti mobili in contesti multicolonna richiede un'attenzione particolare, pena risultati indesiderati. Le seguenti indicazioni dovrebbero risolvere i casi problematici.

Se il documento è composto su due colonne con `twocolumn`:

- si possono inserire oggetti nelle singole colonne mettendoli negli ambienti `table` e `figure` con una larghezza riferita a `\columnwidth` (che è la giustezza della colonna di composizione corrente);
- li si possono inserire a piena pagina mettendoli negli ambienti `table*` e `figure*` con una larghezza riferita a `\textwidth`, ma si noti che nel documento finito compariranno *sempre* in testa alla pagina (le eventuali preferenze indicate, infatti, vengono ignorate) o in una pagina p ;
- si noti che la contemporanea presenza di entrambe le forme degli ambienti altera l'ordine di apparizione degli oggetti nel documento finito: risolve il problema il pacchetto `dblfloatfix`, che permette di indicare le preferenze `[tbp]`.

L'ambiente `multicols`, invece, non è indicato per contenere oggetti mobili. Infatti:

- `table` e `figure` non funzionano e al loro posto vanno usate le forme asteriscate, che si comportano come sopra;
- eventuali oggetti nelle singole colonne si possono solo collocare a mano e "proprio lì" con la preferenza `H` del pacchetto `float` (si veda il paragrafo 6.2.2 a pagina 96).

11.3.5 Colori

Il pacchetto `xcolor` permette di usare i colori in un documento scritto con \LaTeX , anche se in linea generale si sconsiglia di farlo, a meno che non sia davvero indispensabile. Con `xcolor`:

- si possono usare i colori il loro nome, scelto in tavolozze predefinite corrispondenti ad altrettante opzioni;
- si può definire un *qualunque* colore, al quale assegnare un nome arbitrario, secondo svariati modelli;

- si possono definire i colori direttamente nell'argomento dei comandi dedicati via via che servono.

Il funzionamento del pacchetto è piuttosto complicato: se ne veda la ricca documentazione per gli approfondimenti. Di seguito si mostreranno soltanto i comandi più usati e assumendo di usare i colori con il loro nome (quello predefinito oppure quello scelto per un colore personale).

Per **colorare del testo** si usa il comando

```
\textcolor{<colore>}{<testo>}
```

la cui sintassi si spiega da sé (all'interno di un gruppo si può usare la corrispondente dichiarazione `\color`).

Per racchiudere del testo **in un riquadro colorato** nel quale si può determinare la distanza dell'area colorata dal testo, si usa il comando

```
\colorbox{<colore>}{<testo>}
```

tenendo presente che il riquadro modificherà l'interlinea del capoverso e non verrà spezzato su più righe.

Il comando

```
\fcolorbox{<colore_1>}{<colore_2>}{<testo>}
```

racchiude del `<testo>` in un riquadro colorato di `<colore_1>`, del quale si può determinare l'estensione, a propria volta **incorniciato** in una cornice di `<colore_2>`, della quale si può determinare lo spessore. Valgono anche in questo caso le osservazioni fatte per il comando precedente.

Infine, in rari casi potrebbe essere necessario modificare il colore di sfondo di un'intera pagina (come si è fatto nell'ultima pagina di questo capitolo). Per farlo si usa la *dichiarazione*

```
\pagecolor{<colore>}
```

Si noti che si tratta di una dichiarazione *globale*, e dunque metterla in un gruppo non ne limita l'effetto. Risolvono il problema il pacchetto `afterpage` e i seguenti comandi, dati in un punto (meglio tra capoversi "normali") che cadrà nella pagina di cui si vuole cambiare lo sfondo:

```
\pagecolor{<nuovo colore di sfondo>}\afterpage{\pagecolor{white}}
```

In questo modo, dalla pagina successiva lo sfondo ritornerà bianco.

11.3.6 Filigrane

Un semplice modo per proteggere il proprio lavoro ed evitarne usi impropri è quello di contrassegnarne le pagine con una filigrana (come si è fatto in questa pagina). Il pacchetto `draftwatermark`, che agisce anche solo caricandolo, (e il più potente `xwatermark`) permette di farlo facilmente.

11.4 CREARE NUOVI OGGETTI MOBILI

Particolari elementi di un documento come approfondimenti, note storiche o biografiche, esercizi, per esempio, potrebbero creare qualche problema nell'impaginazione, specie se numerosi. Una cura efficace per questo inconveniente è renderli mobili e lasciarne la gestione a \LaTeX , come si è già consigliato per tabelle e figure.

Leslie Lamport, nato il 7 febbraio 1941 a New York City, è l'autore di \LaTeX . Laureatosi in matematica al MIT nel 1960, ha ottenuto un master e un dottorato in matematica alla Brandeis University, rispettivamente nel 1963 e nel 1972. Le ricerche compiute da Lamport hanno gettato le basi della teoria dei sistemi distribuiti. Lamport ha lavorato come informatico al Massachusetts Computer Associates, alla SRI International, alla Digital Equipment Corporation e alla Compaq. Nel 2001 si è unito al gruppo di ricerche di Microsoft, a Mountain View.

Nota biografica 1: Esempio di nuovo oggetto mobile (le notizie riportate sono tratte da Wikipedia).

Il pacchetto `float`, già visto all'opera nel paragrafo 6.2.2 a pagina 96 per forzare la collocazione di un oggetto sulla pagina, permette molto facilmente di creare nuovi oggetti mobili di qualunque tipo, corredandoli di intestazione, numero progressivo e didascalia, isolandoli visivamente dal corpo del testo anche tramite riquadri o filetti ed elencandoli in indici dedicati. La sintassi generale del comando è

```
\newfloat{<nome>}{<preferenze di collocazione>}{<estensione>}[<sezione>]
```

dove

- `\newfloat` (che si comporta come `\newtheorem`) permette di definire una nuova classe di oggetti mobili;
- `<nome>` è il nome scelto per la classe (bio se si tratta di note biografiche, per esempio);
- le `<preferenze di collocazione>` sono le stesse viste nella tabella 30 a pagina 96 per tabelle e figure;
- `<estensione>` è l'estensione scelta per il file ausiliario che \LaTeX userà per produrre l'elenco dei nuovi oggetti (`.lob`, per esempio, che sta per *list of biographies*, "elenco delle note biografiche");
- `<sezione>` specifica a quale tipo di unità di sezionamento (di regola `chapter` o `section`) collegare la numerazione degli oggetti.

La nota biografica contenuta in questa sezione, per esempio, è stata definita nel *preambolo* con il codice seguente, nel quale l'ordine dei comandi va rispettato:

```
\floatstyle{boxed}
\newfloat{bio}{tbp}{lob}
\floatname{bio}{Nota biografica}
```

dove

- `\floatstyle` imposta lo stile della nuova classe di oggetti mobili e prende come argomento uno dei quattro stili definiti dal pacchetto, alla cui documentazione si rimanda il lettore (`boxed`, in questo caso, riquadra l'oggetto e gli mette la didascalia sotto);
- `\floatname` permette di specificare l'etichetta che \LaTeX userà nella didascalia dell'oggetto e la classe a cui assegnarla: in questo caso, *Nota biografica* sarà l'etichetta di tutti gli oggetti mobili della nuova classe `bio` e solo di quelli.

Dopo averli definiti, si potranno inserire i nuovi oggetti nel documento con i consueti comandi d'inizio e fine ambiente. La nota biografica 1 è stata effettivamente inserita nel documento con il codice seguente:

Tabella 57: Parole fisse italiane di babel.

Comando	Voce	Comando	Voce
<code>\abstractname</code>	Sommario	<code>\indexname</code>	Indice analitico
<code>\alsoname</code>	vedi anche	<code>\listfigurename</code>	Elenco delle figure
<code>\appendixname</code>	Appendice	<code>\listtablename</code>	Elenco delle tabelle
<code>\bibname</code>	Bibliografia	<code>\pagename</code>	Pag.
<code>\ccname</code>	e p. c.	<code>\partname</code>	Parte
<code>\chaptername</code>	Capitolo	<code>\prefacename</code>	Prefazione
<code>\contentsname</code>	Indice	<code>\proofname</code>	Dimostrazione
<code>\enclname</code>	Allegati	<code>\refname</code>	Riferimenti bibliografici
<code>\figurename</code>	Figura	<code>\seename</code>	vedi
<code>\glossaryname</code>	Glossario	<code>\tablename</code>	Tabella
<code>\headtoname</code>	Per		

```

\begin{bio}
\textbf{Leslie Lamport}, nato a New York City, è l'autore di \LaTeX.
...
\caption[Esempio di nuovo oggetto mobile]{Esempio di nuovo oggetto
mobile ...}
\label{bio:lamport}
\end{bio}

```

Per produrre l'elenco di tutti gli oggetti di uno stesso tipo, il pacchetto `float` definisce il comando `\listof` (analogo ai comandi già visti per produrre gli elenchi di tabelle e figure) che presenta la sintassi seguente:

```
\listof{⟨nome⟩}{⟨titolo⟩}
```

dove:

- `⟨nome⟩` è lo stesso nome assegnato alla classe dei nuovi oggetti mobili nel primo argomento di `\newfloat`
- `⟨titolo⟩` è il titolo scelto per l'elenco in questione.

Infine, si può uniformare lo stile di tutti gli oggetti mobili del documento (tabelle e figure comprese) con il comando

```
\restylefloat{⟨nome⟩}
```

che ridefinisce lo stile della classe di oggetti mobili indicata con `⟨nome⟩` secondo lo stile indicato con `\floatstyle`. Scrivendo nel preambolo

```

\floatstyle{boxed}
\restylefloat{figure}

```

per esempio, tutte le figure del documento saranno riquadrate.

11.5 ALTRE PERSONALIZZAZIONI

11.5.1 Modificare le parole fisse

Per modificare le parole fisse generate da babel (sostituire *Capitolo* con *Unità* oppure *Bibliografia* con *Opere consultate*, per esempio), si scrive nel preambolo il codice

```
\addto\captions⟨lingua⟩{⟨testo⟩}
```

che comanda a \LaTeX di aggiungere alle definizioni specifiche della $\langle lingua \rangle$ il $\langle testo \rangle$.

Volendo sostituire la voce *Capitolo* con *Unità* basta scrivere

```
\addto\captionsitalian{\renewcommand{\chaptername}{Unità}}
```

mentre per sostituire *Bibliografia* con *Opere consultate* si può scrivere

```
\addto\captionsitalian{\renewcommand{\bibname}{Opere consultate}}
```

La tabella 57 a fronte elenca i comandi di queste voci con la relativa traduzione italiana.

11.5.2 Aggiungere spazio tra le voci dell'indice generale

Se la classe in uso non lo fa già automaticamente, si può rendere la consultazione dell'indice generale più agevole aggiungendo dello spazio tra le voci: di solito lo si mette prima dei titoli di capitolo (come si è fatto in questa guida). Basta caricare il pacchetto `etoolbox` e dare *nel preambolo*:

```
\preto\chapter{\addtocontents{toc}{\protect\addvspace{\langle lunghezza \rangle}}
```

dove $\langle lunghezza \rangle$ va espressa in una qualsiasi delle unità di misura tipografiche accettate da \LaTeX .

11.5.3 Personalizzazioni avanzate

Di seguito si descrivono brevemente alcuni tra i pacchetti più usati per personalizzare in modo avanzato certi aspetti del documento. Si raccomanda di usarli, se proprio necessario, quando il proprio lavoro è in forma *definitiva* e dopo averne letto attentamente la documentazione o la descrizione contenuta in [Gregorio, 2010].

`titlesec`, `titletoc` Permettono di personalizzare ogni aspetto dei titoli di sezione, dell'indice generale, delle testatine e dei piedi.

`sectsty` Permette di cambiare il font dei titoli di sezione e la loro posizione sulla pagina.

`tocloft` Permette di cambiare la resa tipografica degli indici (generale, delle tabelle e delle figure) e di definirne di nuovi.

`tocbibind` Permette di inserire o eliminare i titoli delle sezioni dall'indice generale e di modificarlo in altri aspetti.

`fancyhdr` Permette di personalizzare ogni aspetto di testatine e piedi.

`enumitem` Permette di personalizzare ogni aspetto dei tre ambienti standard per gli elenchi `itemize`, `enumerate` e `description`.

Come si è detto più volte, anche se \LaTeX induce a privilegiare la struttura logica del documento e a trascurarne l'aspetto finale, nemmeno lui, tuttavia, riesce a risolvere in una volta sola *tutti* i problemi tipografici evitando d'intervenire in prima persona: è il piccolo prezzo da pagare per un prodotto di altissima qualità. La revisione finale del documento è una fase delicata: si tratta, a volte, di risolvere grandi problemi d'impaginazione con piccoli (o piccolissimi) aggiustamenti. Un'arte complicata ma ricca di soddisfazioni, che dà i propri frutti migliori solo quando il documento è nella sua forma *definitiva*.

Prima di mettere in pratica i suggerimenti proposti nei prossimi paragrafi si tenga presente che *quasi* sempre si può correggere un difetto d'impaginazione semplicemente riformulando il testo nel punto o nei dintorni del punto critico. Si risolva un problema alla volta, un capoverso alla volta: *probabilmente* anche qualche altro problema più in là nel testo andrà a posto da solo.

12.1 PROBLEMI ORIZZONTALI

I difetti d'impaginazione orizzontali riguardano la formazione dei capoversi: di solito consistono in righe sporgenti nel margine destro.

Titoli problematici

Se troppo lunghi, i titoli creano problemi almeno in due luoghi:

- nel corpo del documento, perché \LaTeX potrebbe spezzarli in modo non soddisfacente o addirittura non spezzarli del tutto se non trova un punto di sillabazione adatto;
- nell'indice generale e nelle eventuali testatine, riproducendo il problema appena visto.

Un titolo di sezione non dovrebbe *mai* andare a capo. Si risolvono tutti i problemi in una volta sola riformulandolo: si può fare praticamente sempre. Se invece il titolo lungo fosse davvero necessario, lo si mandi a capo con un `\` esplicito nell'argomento del comando di sezionamento e se ne usi l'argomento facoltativo per indice generale e testatine:

```
\chapter[Titolo breve per indice e testatine]%
  {Titolo lungo\ da mandare a capo}
```

Capoversi problematici

Un capoverso che \LaTeX non riesce a comporre bene può influenzare la divisione in pagine del documento, per cui sistemarlo *potrebbe* risolvere automaticamente anche alcuni dei problemi descritti nel prossimo paragrafo. Di seguito si danno alcuni consigli per le situazioni più frequenti.

Un documento di una certa lunghezza conterrà quasi certamente qualche riga che \LaTeX non è riuscito a comporre “bene” e che si vedrà sporgere (ma non sempre, si noti) nel margine destro. Per essere davvero sicuri di “curarle” tutte, si possono evidenziare dando immediatamente prima dell’inizio del documento il comando

```
\overfullrule=<lunghezza>
```

che stampa un \blacksquare accanto a ciascuna (più meno grande a seconda del valore assegnato a $\langle\text{lunghezza}\rangle$, esprimibile in una qualunque delle unità di misura tipografiche accettate da \LaTeX).

Possono causare lo stesso problema anche le parole che \LaTeX per vari motivi non riesce a spezzare e che dopo la composizione si trovano a fine riga: si provi a suggerirne la sillabazione con degli \backslash -espliciti là dove serve.

Si possono risolvere altre situazioni problematiche eliminando qualche spazio indivisibile non necessario (*mai*, però, tra l’iniziale puntata di un nome e il relativo cognome) o mettendo in display alcune formule matematiche poco leggibili in linea. Infine, si usino il meno possibile lunghi URL nel corpo del testo: li si metta in una nota al piede (se sono pochi ed è proprio necessario) oppure, se molti, in un elenco alla fine del documento.

Gli ultimi tocchi al documento sono puramente estetici. Si eviti che due righe consecutive comincino o finiscano con la stessa o le stesse parole riformulando il testo; si eviti che un capoverso termini con una sola parola molto breve (o, peggio, con una sola sillaba) racchiudendola in $\backslash\text{mbox}$ e unendola alla penultima con uno spazio indivisibile. Lo spazio sottile $\,$, è preziosissimo per risolvere altri inestetismi: può servire per separare un apostrofo e le successive virgolette inglesi; per evitare fastidiose sovrapposizioni in certe sequenze: tra una lettera corsiva con tratti discendenti come la *f* e una parentesi tonda immediatamente prima, per esempio, oppure tra l’esponente di una nota al piede e ciò che lo precede.

12.2 PROBLEMI VERTICALI

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

Oggetti in testo e fuori testo

Gli oggetti in testo causano problemi d’impaginazione a volte irrisolvibili perché, di regola, non possono essere spezzati *in nessun modo* tra due pagine. Basta evitarli il più possibile e non si avranno di questi problemi.

Gli oggetti fuori testo, invece, sono molto più flessibili. Tenendo presente che si può considerare “ottimale” il risultato fintanto che oggetto e relativo riferimento si trovano sulla stessa pagina o al massimo in due pagine opposte (a libro aperto non dovrebbe essere necessario voltare pagina, per intenderci), anche se purtroppo non sempre è possibile, qualche oggetto potrebbe comunque non piacere dove \LaTeX ha pensato di metterlo:

- potrebbe essere finito un po’ troppo lontano dal punto “ottimale”;
- potrebbe essere finito in una pagina insieme a troppi titoli di sezione, ciò che renderebbe la pagina troppo vuota;
- ci si potrebbe trovare con due oggetti in una pagina e nessuno in quella successiva, quando li si preferirebbe distribuiti più omogeneamente;

- si potrebbe volere in una pagina di soli oggetti un oggetto che invece \LaTeX ha messo in una pagina con del testo.

Come fare? I primi tre casi si risolvono semplicemente arretrando il codice dell'oggetto di qualche capoverso, mentre il quarto dandogli la preferenza p , che \LaTeX riesce a soddisfare *sempre* anche se da sola. Si possono sbrogliare tutte le altre situazioni intricate usando oculatamente le preferenze di collocazione e facendo qualche prova, considerando che molto testo e pochi oggetti garantiscono risultati migliori della situazione contraria.

Si danno, infine, ulteriori suggerimenti utili:

- a volte può risolvere la situazione `\clearpage`, spiegato nel paragrafo 12.2 a pagina 189;
- si può provare con `\FloatBarrier` (richiede il pacchetto `placeins`), che traccia delle barriere invalicabili per gli oggetti mobili;

Note a piè di pagina

Le note al piede sono problematiche specie se lunghe più di un capoverso, perché sono ottime candidate a finire tra due pagine moltiplicando i difetti. Le si eviti il più possibile, dunque.

Orfani e vedove nel corpo del documento

In tipografia si usa chiamare *orfano* la prima e unica riga di un capoverso in fondo alla pagina e *vedova* l'ultima riga di un capoverso in cima a una pagina nuova. Che siano gli orrori tipografici da evitare con la cura più certolina lo conferma la terminologia tedesca: una riga orfana si chiama *Schusterjunge* ("apprendista ciabattino"), mentre una riga vedova *Hurenkind* ("figlio di p***"). Bisogna fare in modo che alle due estremità di una pagina ci siano sempre *almeno due* righe di uno stesso capoverso, e almeno quattro nell'ultima pagina di un capitolo. È accettabile, invece, una riga solitaria in cima alla pagina immediatamente seguita da testo in display di qualunque tipo. \LaTeX è programmato per evitare automaticamente queste due situazioni: quando non ce la fa, si ricorra ai consigli descritti di seguito.

Il metodo più semplice per risolvere la faccenda è riformulare il capoverso in questione per diminuirlo o aumentarlo di una riga (funziona in entrambi i casi) oppure valutare la possibilità di spezzare un capoverso in due o di riunirne due in uno nei pressi del problema.

Se il capoverso che contiene l'orfano o la vedova è abbastanza lungo, si può provare a dare *immediatamente prima o dopo* il testo del capoverso (o anche in mezzo) uno dei due comandi

```
\looseness=1
\looseness=-1
```

con cui si chiede a \LaTeX di provare ad allungarlo o accorciarlo del numero di righe indicato (che è 0 per impostazione predefinita). Un tentativo con -1 potrebbe funzionare se l'ultima riga è *molto* breve, uno con 1 se è quasi piena. Si ricordi che il comando agisce solo sul capoverso cui è applicato.

Si possono eliminare orfani e vedove anche spostando una riga di testo dalla pagina *precedente* a quella in cui si trovano con `\pagebreak`. Il comando va dato prima della riga orfana nella versione composta e funziona tanto meglio quanto più la pagina contiene spazi flessibili. Lo si usi senz'altro quando un display sta in cima alla pagina e il capoverso precedente è lungo una o due righe soltanto. Si ricorda anche l'opzione `noorphans` dell'ambiente `quoting`.

Se nessuno degli strumenti visti fin qui funziona, si può provare con uno tra

```
\enlargethispage{1\baselineskip}
\enlargethispage{-1\baselineskip}
```

che dicono a L^AT_EX di allungare o accorciare *la pagina* del numero di righe indicato nell'argomento: è bene che l'allungamento (o l'accorciamento) ammonti a una riga soltanto o, *in casi particolarissimi*, a due, ma non di più. Il comando *modifica la gabbia del testo* e va dato *tra due capoversi*: perciò *va usato in coppia su due pagine opposte*, il cui piede non sia occupato da note o numeri di pagina e con margini inferiori sufficientemente alti. Se per la stampa si userà una carta di qualità, infine, l'aggiustamento non si noterà nemmeno. La variante asterisco del comando si comporta allo stesso modo, provando prima a comprimere o ad allargare gli eventuali spazi bianchi sulla pagina a seconda dei casi.

Si ricordi, infine, che salvo situazioni fortunate i capoversi brevi non sono adatti per simili acrobazie.

Orfani e vedove nell'indice generale

Il problema appena descritto può affliggere anche le voci dell'indice generale del documento.

Per riportare all'ovile una voce vedova si può dare

```
\addtocontents{toc}{\protect\enlargethispage*{\baselineskip}}
```

nel corpo del documento *immediatamente prima* di un titolo di sezione che nella versione composta del documento cade nella pagina dell'indice generale precedente la vedova. Si facciano delle prove aumentando il valore di `\baselineskip` finché la voce in questione non rientra.

Per dare un po' di compagnia a una voce orfana, invece, si darà

```
\addtocontents{toc}{\protect\pagebreak}
```

nel corpo del documento *immediatamente prima* del titolo di sezione incriminato, tenendo presente che la pagina dell'indice verrà stiracchiata con tutte le conseguenze del caso.

Pagine di solo testo

I capoversi più problematici per il buon riempimento della pagina sono quelli di solo testo, specie se lunghi, numerosi e contigui. Infatti, gli eventuali spazi aggiunti dal programma visti nel paragrafo 3.6.2 a pagina 32 cadrebbero inevitabilmente tra un capoverso e l'altro, risultando immediatamente (e fastidiosamente) visibili. Si risolve il problema in uno dei modi seguenti:

- riformulando il testo in uno o più capoversi;
- inserendo nella pagina un oggetto mobile o uno o più display di qualunque tipo;
- facendo in modo che nella pagina ci sia almeno un titolo di sezione.

Gli ultimi due rimedi, in particolare, permettono al programma di ripartire lo spazio in più in modo ottimale.

Interrompere la pagina corrente

Di seguito si descrivono brevemente i principali comandi standard per cambiare pagina. Tutti interrompono la pagina corrente *nel punto in cui vengono dati*, ma:

- `\newpage` comincia semplicemente una pagina nuova;
- `\pagebreak` prima di cominciare la nuova pagina stiracchia in verticale il contenuto di quella in cui viene dato per riempirla al meglio;
- `\clearpage` prima di cominciare la nuova pagina dice a \LaTeX di stampare tutti gli oggetti già definiti e che non hanno ancora trovato posto sulle pagine, se ce ne sono (si veda il paragrafo 6.2.2 a pagina 93);
- `\cleardoublepage` si comporta come il comando precedente ma inserendo, se necessario, una pagina bianca prima di cominciare quella nuova (utile nei documenti impostati per la stampa in fronte/retro).

Si tenga *ben* presente che è \LaTeX a interrompere automaticamente la pagina nel punto ritenuto migliore: i comandi appena descritti ne modificano il comportamento predefinito e possono risolvere situazioni problematiche, ma vanno usati con grande accortezza.

Controllare l'indice analitico

Per controllare l'indice analitico può essere utile il pacchetto `showidx`, che ne visualizza le voci nel margine sinistro delle pagine del documento.

E per finire...

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

A

NORME TIPOGRAFICHE

Quest'appendice, basata su [Cevolani, 2006] e [Lesina, 2004], cui si rimanda per gli approfondimenti, descrive sinteticamente e senza pretese di completezza le tradizioni tipografiche più seguite nella redazione di un documento in italiano. Di ogni regola discussa si mostra, per quanto possibile, l'applicazione in \LaTeX . La parola *norma* va qui intesa in senso piuttosto lato: anche nella nostra lingua, come in tutte le altre, non esistono che pochissime regole tipografiche realmente universali e vincolanti, mentre molti aspetti del testo finito dipendono da tradizioni e abitudini o dai gusti di utente o editore.

Ciò premesso, si tenga presente che la scrittura in una qualunque lingua è governata nel mondo dalle norme emesse dall'ISO e in Italia da quelle definite dall'UNI, che hanno valore di legge. Per maggiori dettagli su queste norme, se ne veda la documentazione.

Naturalmente è superfluo ricordare che per risolvere dubbi di altro tipo il mezzo più rapido e sicuro rimane la consultazione di grammatica e dizionario.

A.1 ACCENTO E APOSTROFO

Accento

In italiano ci sono due accenti:

- *grave*, ` , che indica suono aperto;
- *acuto*, ´ , che indica suono chiuso.

Un terzo accento, il *circonflesso*, ^ , usato in passato soprattutto per distinguere gli omografi uscenti in *-ii* al plurale – *odî* (plurale di *odio*) da *odi* (voce del verbo *udire*) – oggi non viene quasi più usato.

La loro applicazione non è univoca, e oscilla tra le prescrizioni UNI (per cui l'accento grave può colpire *tutte* le vocali: à/À, è/È, ì/Ì, ò/Ò, ù/Ù; e l'accento acuto soltanto le vocali *e* e *o*: é/É, ó/Ó) e le tradizioni di alcune case editrici (per cui *i* e *u* portano sempre l'accento acuto, per esempio). Qualunque sistema si scelga, l'importante è seguirlo coerentemente in *tutto* il documento.

Come si può osservare, gli accenti vanno apposti *sopra* la vocale, minuscola o maiuscola che sia: il paragrafo 3.2.1 a pagina 19 mostra come impostare i pacchetti che permettono di interpretare correttamente le lettere accentate italiane. Si noti tuttavia che per ottenere alcuni caratteri non presenti sulla tastiera italiana come la *ó* bisogna sempre digitare il codice esplicito o le relative combinazioni di tasti. Al contrario di quanto accade con le vecchie macchine per scrivere o con altri elaboratori di testo, \LaTeX permette di accentare senza problemi *anche le lettere maiuscole*, che spesso vengono erroneamente apostrofate (come in *E'*, che sta per *Ei* cioè "egli").

Apostrofo

L'*apostrofo*, *'*, segnala normalmente la caduta della parte iniziale, come in *'sta* (per *questa*), o finale di una parola, come in *un'altra*, *un po'*, *da'* (imperativo), eccetera.

Quando cadono lettere o sillabe iniziali l'apostrofo è preceduto da uno spazio; quando cadono sillabe finali l'apostrofo è seguito da uno spazio o da un segno d'interpunzione.

Quando la vocale finale di una parola cade di fronte a quella iniziale della parola successiva (come in *un'altra* o in *quell'uomo*), l'apostrofo sta al posto della vocale caduta e la parola seguente comincia senza nessuno spazio intermedio. Se babel è caricato con l'opzione *italian*, L^AT_EX evita automaticamente l'apostrofo in fine di riga: *quell'uomo* viene sillabato "quel-l'uo-mo", come se fosse un'unica parola (con altre lingue che usano l'apostrofo, però, questo potrebbe non succedere).

A.2 PUNTEGGIATURA E SPAZIATURA

La punteggiatura italiana comprende segni d'interpunzione, parentesi, virgolette, puntini di sospensione, trattini e altri simboli come asterisco e barra. Ci sono alcune regole fisse sull'uso degli spazi prima e dopo i segni di punteggiatura.

A.2.1 Segni d'interpunzione

Tutti i segni d'interpunzione seguono immediatamente la parola precedente e vanno separati con uno spazio da quella successiva.

A.2.2 Virgolette

Nel testo di regola non si lasciano spazi bianchi tra virgolette e contenuto (anche se è ammesso e talvolta richiesto lo spazio sottile), mentre li si lascia tra virgolette e testo esterno.

Normalmente si evidenziano tra virgolette parole o frasi in relazione al loro significato:

- citazioni dirette e citazioni "annidate", cioè citazioni dentro a citazioni («Gli ho sentito dire: "Verrò di sicuro"»);
- termini ed espressioni che specificano il significato di altre parole: «La parola *box* significa "scatola"»;
- espressioni figurate o gergali: «Si prevedono scioperi "a singhiozzo"»;
- termini correnti a cui si attribuisce un significato diverso da quello abituale: «Questo ragazzo non è certo una "cima"»;
- parole usate in senso ironico.

I casi sopra elencati non sono regole vincolanti: alcuni di essi, infatti, trovano un ottimo concorrente nel *corsivo*, con l'avvertenza di evitare *sempre* di usarli contemporaneamente. Non esistono nemmeno regole fisse o comunemente accettate per usare i vari tipi di virgolette (si veda il paragrafo [4.3.1](#) a pagina [52](#)): nella pubblicazione dei testi a stampa, infatti, si adottano convenzioni tipografiche variabili dall'uno all'altro editore e perfino tra l'una e l'altra collana dello stesso marchio editoriale. L'unico consiglio che qui si

può dare è di scegliere un sistema di virgolettatura *prima* di cominciare a scrivere il documento e di usarlo con coerenza.

A.2.3 Parentesi

In italiano si usano di solito le parentesi *tonde* () e le parentesi *quadre* []. Le parentesi *graffe* { } e *angolate* < > (i comandi per queste ultime sono `\langle` e `\rangle`, da usare in modo matematico) si usano solo in matematica e in informatica rispettivamente.

In genere le parentesi racchiudono un inciso nel discorso (cioè una parte *che si può omettere*, come questa). Seguono alcune indicazioni generali sul loro impiego.

- Le parentesi vanno attaccate al loro contenuto e separate con uno spazio dal testo precedente e seguente (a meno che non chiudano un enunciato, come qui).
- Quando un segno d'interpunzione chiude il testo tra parentesi, si richiede, come qui, un ulteriore punto *fuori* dalla parentesi di chiusura (ovviamente!). Si noti che di norma solo i punti interrogativo ed esclamativo possono stare dentro le parentesi.
- L'uso di racchiudere tra parentesi un intero enunciato è lecito ma poco diffuso in italiano, a differenza di altre lingue. (In tal caso, anche la punteggiatura rientra nelle parentesi, come qui.)

Le parentesi quadre si usano quasi esclusivamente in due casi.

- Come parentesi interne a parentesi (come [anche se non è molto frequente] in questo caso).
- Per introdurre «il commento di una persona diversa dall'autore del testo cui il commento si riferisce [in questo modo]». In quest'ultimo caso rientra anche quello dell'*omissione* volontaria (che è comunque un commento), segnalata con [...] (si veda il paragrafo 4.3.1 a pagina 52).

A.2.4 Puntini di sospensione

I puntini di sospensione sono *sempre e solo tre* e, come gli altri segni di interpunzione, seguono immediatamente la parola che li precede e sono separati con uno spazio da quella che li segue... in questo modo. Con \LaTeX , il comando da usare è `\dots` (non vanno *mai* inseriti a mano tre punti separati), eventualmente chiudendolo con un gruppo vuoto{} se i puntini non si trovano a fine enunciato (si veda il paragrafo 4.3.1 a pagina 52).

Se usati per indicare un'omissione in una citazione «è bene [...] inserirli tra parentesi quadre o tonde» con il comando `\omissis` come in questo caso (si veda lo stesso paragrafo).

A.2.5 Trattino, tratto e lineetta

Si è già spiegato (nel paragrafo 4.3.1 a pagina 52) come realizzare questi tre segni con \LaTeX . Di seguito se ne descrive l'uso in termini generali.

Il *trattino* di solito divide le parti di un'espressione composta e nella scrittura s'interpone tra di esse senza ulteriori spazi bianchi. Si usa:

- Nei termini formati da due parole autonome, come «guerra-lampo».

- Negli intervalli numerici o di tempo i cui estremi siano espressi in cifre, come «1915-1918» e «pagine 2-11». Se gli estremi sono costituiti da più di una parola, per maggiore chiarezza è opportuno spaziare anche il trattino prima e dopo: «Trentino - Alto Adige».

Il *tratto*, preceduto e separato da uno spazio bianco, di solito suddivide elementi come titoli, didascalie e diciture di vario tipo, come «Varianti di carattere – Il maiuscoletto».

La *lineetta* isola nettamente un inciso all'interno del testo — ma in quest'uso la tipografia italiana preferisce il tratto — o segnala le battute in un dialogo. Nella scrittura è separata dal testo precedente e seguente da uno spazio bianco normale, come qui.

A.2.6 Barra e asterisco

La *barra* (o *sbarretta*) può essere usata nelle seguenti circostanze ordinarie.

- Senza spazi né prima né dopo, per un'alternativa tra due termini: «i passeggeri diretti a Torino/Milano».
- Preceduta e seguita da uno spazio bianco, separa i versi di una poesia in linea: «M'illumino / d'immenso».
- Senza spazi prima e dopo, si usa nelle frazioni numeriche in modo testuale: «i 3/4 della popolazione».

Si ricordi che la barra (/), inseribile direttamente dalla tastiera, *non* va confusa con il carattere di barra rovescia (\), riservato ai comandi di \LaTeX .

L'*asterisco* ha un uso limitato alle seguenti situazioni:

- separato da quanto precede con uno spazio sottile e ripetuto tre volte, indica omissione volontaria («Il padre Cristoforo da *** era un uomo più vicino ai sessanta che ai cinquanta», codice: *\, *\, *);
- per indicare in linguistica forme non attestate, scorrette o inaccettabili: «* che io vadi».

A.3 STILE DEL FONT

Oltre a quello normale usato per il testo, di solito un font prevede lo stile *corsivo*, **nero** e MAIUSCOLETTA. Si usano poco lo stile *inclinato* (da non confondere con il corsivo *inclinato*) e altri stili, utili solo per esigenze particolari: per esempio, in questo documento si usa lo stile dattilografico per evidenziare il codice \LaTeX . Lo stile “evidenziato” (o “enfattizzato”) è reso normalmente con il corsivo, ma riveste un ruolo logico differente.

A.3.1 Corsivo

Normalmente si evidenziano in corsivo parole o frasi in relazione alla loro presenza nel testo:

- termini tecnici e specialistici: «Una *distribuzione* è una raccolta di programmi...»;
- parole o frasi straniere di uso non comune con le quali si ritiene che il lettore non abbia affinità: «Questa tecnica di *engraving* può essere considerata...»;

- parole e lettere a cui ci si riferisce come tali nel testo: «La lettera *e* non compare nella parola *parola*»;
- parole o frasi da evidenziare: «Questo *non* si fa».

Oltre che nel testo principale, si può usare il corsivo per comporre i titoli.

A.3.2 Nero

Le classi standard di L^AT_EX usano il nero per i titoli di sezione. Di norma non va impiegato per evidenziare parole o interi capoversi del testo principale (il corsivo va benissimo), e comunque va utilizzata con moderazione per non appesantire la pagina.

A.3.3 Maiuscoletto

Si usa il maiuscoletto quasi esclusivamente per i nomi degli autori citati in bibliografia, come in BRINGHURST (1992). Questa convenzione, tuttavia, dipende dallo stile bibliografico scelto. Oppure concorre con il maiuscolo per segnalare gli acronimi nel testo.

A.4 TRATTAMENTO DEL TESTO

Per una trattazione completa delle varie tipologie di parole si rimanda a [Lesina, 2004]. Come esempio, si prenderà in considerazione il caso delle parole straniere, uno dei più frequenti.

A.4.1 Parole straniere

Le parole straniere vanno in corsivo, a meno che non vengano esplicitamente “quoted” (“virgolettate”, come in questo caso) o non siano d’uso comune. Quindi si scriverà «ho visto un bel film» (comune) ma «ho mangiato un *pudding*» (non comune). In realtà, essendo molto difficile stabilire che cosa sia “comune” o meno, la cosa più corretta è scrivere in corsivo le parole straniere che si presumono di uso non corrente *per il lettore a cui ci si rivolge*. In un libro d’informatica, perciò, *software* e *computer* saranno in tondo.

I nomi propri e le denominazioni ufficiali (come Stanford University e Magna Charta) non sono considerati parole straniere e vanno in tondo.

La traduzione straniera di un’espressione italiana usata nel testo può essere semplicemente messa in corsivo e fra parentesi (*bracket*), come in questo caso. Se invece l’espressione tradotta ricorre in una citazione e se ne vuole indicare la forma originale, va tra parentesi quadre come ogni altro commento. Per esempio: «La visione del mondo [*Weltanschauung*]. . . ».

Ovviamente ogni lingua straniera va sillabata a sé, ed è altrettanto ovvio che scrivendo in lingue diverse dall’italiano alcune cesure nel documento finito potrebbero risultare errate. Si noti che è ammesso sillabare secondo le nostre regole parole straniere isolate nel discorso, ma non un testo in lingua più esteso (si veda il paragrafo 3.2.1 a pagina 19).

A.4.2 Numeri

Scrivere i numeri

I caratteri numerici possono essere *maiuscoli* e *minuscoli*. Si osservi la resa tipografica del numero 1821 nei due esempi seguenti:

Il numero $\$1821\$$ non è primo. \\ Napoleone morì nel 1821.	Il numero 1821 non è primo. \\ Napoleone morì nel 1821.
---	---

Nel primo, 1821 è maiuscolo; nel secondo, 1821 è minuscolo.

I numeri maiuscoli hanno tutti la stessa altezza e sono indicati:

- nelle formule matematiche;
- nelle tabelle composte di dati numerici;
- se intesi in senso aritmetico (come nell'indicazione di quantità esatte).

Mentre le prime due situazioni si curano con gli strumenti già esaminati nei capitoli precedenti, l'ultima richiede di scrivere i numeri in uno dei due modi seguenti:

- tra i comandi per le formule matematiche in linea ($\$(numero)\$$);
- nell'argomento del comando `\num` del pacchetto `siunitx`.

I numeri minuscoli presentano tratti ascendenti e discendenti per integrarsi al meglio con i caratteri minuscoli del font corrente, e sono indicati in tutti gli altri casi. Di solito per ottenerli non serve fare nulla, perché spesso sono predefiniti nel font scelto: si consiglia in ogni caso di fare una semplice prova per verificare la cosa e di consultare la documentazione del pacchetto per invertire eventualmente la situazione.

Si noti, infine, che anche se alcuni font (come quelli predefiniti di \LaTeX) non distinguono tra numeri maiuscoli e minuscoli, si dovrebbe assumere da subito la buona abitudine di usare i dollari nei casi previsti: così facendo si potrà cambiare il font del documento senza dover modificare a mano ogni numero.

Spaziare le cifre

La corretta scrittura dei numeri *di cinque o più cifre* prevede uno spazio sottile ogni tre cifre a partire da destra (come in 1 500 000). Per ottenerlo ci sono diverse possibilità: se i numeri da scrivere nel documento non sono molti, si può inserire a mano lo spazio sottile `\,`; in caso contrario, risolve il problema il comando `\num` visto nel paragrafo precedente. Si osservi il risultato tipografico dei numeri negli esempi seguenti:

1500000 \\	1500000
\$1500000\$ \\	1500000
1\,500\,000 \\	1 500 000
\$1\,500\,000\$ \\	1 500 000
<code>\num{1500000}</code>	1 500 000

Come si può notare, se non altrimenti specificato `\num` produce numeri maiuscoli.

Si evitino il punto e la virgola per separare le cifre, perché questi due segni servono già da separatore decimale nel mondo anglosassone ed europeo rispettivamente. Lo spazio sottile è l'*unico* metodo universalmente corretto.

Numeri decimali

Nell'uso italiano, il separatore fra parte intera e fratta di un numero decimale è una virgola: «21,12». Si noti che in modo matematico \LaTeX considera la virgola un normale segno di punteggiatura e mette dopo uno spazio extra, con un risultato insoddisfacente. Il pacchetto `siunitx` caricato con l'opzione `output-decimal-marker={,}` e `\num` risolvono anche questo problema:

<code>\$21,12\$ \ \</code>	21,12
<code>\num{21,12}</code>	21,12

Si osservi che nel primo caso dopo la virgola si ha una spaziatura (leggermente) sbagliata: la scrittura corretta è la seconda.

A.4.3 Frazioni, percentuali, unità di misura

Le frazioni si esprimono in lettere, «tre quarti», a meno che non indichino una quantità numerica precisa. In questo caso si possono scrivere usando la barra, «3/4», o la forma frazionaria vera e propria, « $\frac{3}{4}$ », ottenibile con il codice visto nel paragrafo 5.2.3 a pagina 68.

Nella scrittura del testo il simbolo di percento segue immediatamente il numero cui si riferisce senza spazi intermedi: «30%» (codice: `\%`; attenzione ai caratteri speciali).

Le *quantità misurate* sono costituite da numeri seguiti da un'unità di misura espressa di solito con il simbolo relativo, come in «20 cm» o «15 kg». Si noti che tra numero simbolo c'è uno spazio sottile e che il simbolo non vuole il punto dopo. Per scrivere correttamente le unità di misura del Sistema Internazionale si consiglia di usare il pacchetto `siunitx` o lo spazio sottile (si veda il paragrafo 5.12 a pagina 89).

A.4.4 Acronimi e abbreviazioni

Gli *acronimi* sono espressioni formate dalle lettere o sillabe iniziali delle parole di un'espressione che si vuole abbreviare, come `GUT`, HTML o PDF. Dovrebbero essere composti interamente da maiuscole (si ammette anche il maiuscoletto), *non* spaziate e *senza* punti d'abbreviazione. Se l'acronimo è entrato nell'uso e si può pronunciare come parola, è ammissibile (e spesso preferibile) scriverlo come tale: *Fiat* e *radar* (anziché FIAT e RADAR) vanno benissimo. Si consiglia di citare per esteso gli acronimi meno noti la prima volta che compaiono nel testo, mettendone la forma estesa fra parentesi: `GUT` (Gruppo Utilizzatori Italiani di `TEX` e `LATEX`).

Le *abbreviazioni* si ottengono invece dal troncamento di una parola mantenendone una o più lettere iniziali seguite dal punto. In casi particolari, come *sig.ra* o *prof.ssa*, l'abbreviazione comprende anche la parte finale della parola originale, che segue il punto senza spazi in mezzo. Se l'abbreviazione dovesse cadere a fine enunciato (caso raro, forse possibile con il solo *ecc.*) il punto di abbreviazione funziona anche da punto fermo (cioè, ovviamente, non si scrivono due punti successivi).

quantità, si possono usare i comandi standard mostrati nella tabella 17 a pagina 54. In entrambi i casi \LaTeX potrebbe sillabare in modo insoddisfacente o non sillabare per nulla le parole con la quantità segnata: si risolve il problema indicando esplicitamente i punti di cesura con "|" là dove serve.

Si noti che non è permesso (né avrebbe senso) indicare la quantità di un dittongo legato in latino medievale (si veda il paragrafo seguente).

B.1.2 Latino medievale

Per dare al testo latino una patina medievaleggiante, l'utente può servirsi delle seguenti varianti (tutte o solo alcune):

- usare un unico segno, *u*, per *u* e *v* minuscole: *Nouembris* anziché *Novembris*, per esempio;
- usare un unico segno, *V*, per *U* e *V* maiuscole: *IVLIVS* anziché *Iulius*;
- usare le legature per i dittonghi *æ* e *œ* (minuscoli e maiuscoli), che nel latino classico erano *sempr*e sciolti: *æ* (`\ae`), *œ* (`\oe`), *Æ* (`\AE`), *Œ* (`\OE`).

L'opzione `latin` di `babel` definisce un paio di automatismi utili per scrivere in questa varietà di latino:

- gestisce correttamente la conversione da minuscolo a maiuscolo nelle stringhe automatiche come le testatine: un titolo come `\chapter{C\ae sar et Heluetii}`, per esempio, nella corrispondente testatina maiuscola diventa *CÆSAR ET HELVETII*;
- ridefinisce `\prefacename` per ottenere *Præfatio* anziché *Praefatio* (le altre parole fisse non subiscono variazioni, si veda il paragrafo seguente).

Si seleziona questo tipo di scrittura caricando l'attributo `medieval` subito dopo la chiamata di `babel`:

```
\languageattribute{latin}{medieval}
```

Anche usando l'ortografia medievale si possono risolvere *eventuali* sillabazioni problematiche come già spiegato nel paragrafo precedente.

B.1.3 Documenti interamente in latino

Chi dovesse scrivere un documento completamente in latino può contare su qualche agevolazione in più. L'opzione `latin` di `babel`, infatti:

- converte automaticamente la data prodotta da `\today` nella corrispondente data "latina", nella forma *XXIII Novembris MMXII*;
- traduce in latino tutte le parole fisse delle classi di documento standard: *Capitolo* diventa *Caput*, per esempio.

Si fornisce infine il preambolo tipo per comporre un documento in italiano e latino o nel solo latino (le righe sono numerate per maggiore chiarezza).

```
1 \documentclass[a4paper]{article}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4
5 \usepackage[latin,italian]{babel}
6 \usepackage[latin]{babel}
7
8 \languageattribute{latin}{withprosodicmarks}
```

```

9 \languageattribute{latin}{medieval}
10
11 \begin{document}
12
13 ...
14
15 \end{document}

```

Si noti che:

- va indicata obbligatoriamente *una sola* tra le righe 5 e 6 a seconda della o delle lingue del documento;
- le righe 8 e 9, invece, servono soltanto per sfruttare le rispettive specialità di `latin` e si possono anche caricare insieme.

B.2 SCRIVERE IN GRECO

Scrivere in greco con \LaTeX richiede qualche attenzione: non soltanto è diverso l'alfabeto, ma lo sono anche la codifica di output e i font usati nel documento finito.

Prima di mettersi a scrivere, bisogna scegliere almeno due cose:

1. il *sistema ortografico* che s'intende seguire;
2. il *metodo di scrittura* del testo greco nel sorgente.

\LaTeX gestisce correttamente due sistemi ortografici per scrivere in questa lingua:

- il sistema *politonico* ("con molte intonazioni"), comprendente tre accenti e numerosi altri segni diacritici, con cui si può scrivere il greco antico;
- il sistema *monotonico* ("con una sola intonazione"), comprendente un solo accento, con cui si può scrivere il greco moderno.

Nel sorgente, infine, si possono scrivere le parti in greco in due modi:

- *traslitterando* il testo greco secondo una particolare corrispondenza tra i caratteri;
- scrivendo *direttamente* i caratteri greci.

Le possibilità appena descritte si possono combinare per un totale di quattro. Se non diversamente indicato, gli esempi di questo paragrafo sono in greco antico politonico.

L'opzione `greek` tra le lingue di `babel`

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

carica automaticamente i font greci predefiniti CB (di Claudio Beccari, compresi nelle moderne distribuzioni di \LaTeX , ma ne sono disponibili molti altri) e attiva la codifica di output LGR (*Local Greek*) e la corretta sillabazione delle parole, ma permette di scrivere solo in greco moderno. Si noti che per la traduzione delle parole fisse in greco antico in un documento interamente in greco (si veda il paragrafo [B.2.3](#) a pagina [206](#)) e la corretta interpretazione del carattere `~` (si veda il paragrafo [B.2.1](#) nella pagina seguente) l'opzione da specificare è `polutonikogreek`. Si passa da una lingua all'altra con gli strumenti già esaminati nel paragrafo [3.2.1](#) a pagina [19](#) come, per esempio:

	.	!	"	£	\$	%	&	/	()	=	?	;	^									
\	1	2	3	4	5	6	7	8	9	0	'	'	ì										
Q	X	W	Ω	E	E	R	P	T	T	Y	Ψ	U	Υ	I	I	O	O	P	Π	é	*		
q	x	w	ω	e	ε	r	ρ	t	τ	y	ψ	u	υ	i	ι	o	ο	p	π	è	[+]
A	A	S	Σ	D	Δ	F	Φ	G	Γ	H	Η	J	Θ	K	Κ	L	Λ	ç	°	§			
a	α	s	σ	d	δ	f	φ	g	γ	h	η	j	θ	k	κ	l	λ	ò	@	à	#	ù	
>	'	Z	Z	X	Ξ	C		V		B	B	N	N	M	M	;	.	:		-			
<	°	z	ζ	x	ξ	c	ς	v		b	β	n	ν	m	μ	,	.	:		-			

Figura 17: Corrispondenza tra la tastiera italiana e i caratteri greci.

`\dots` testo in italiano.

```
\begin{otherlanguage*}{greek}
```

```
Th| p'anta dido'ush| ka'i
>apolambano'ush| f'usei <o
pepaideum'enos ka'i a>id'hmwn
l'egei; <<d'os, <'o j'eleis,
>ap'olabe, <'o j'eleic>>.
L'egei d'e to-uto o>u
katajrasun'omenos, >all'a
peijarq~wn m'onon ka'i
e>uno~wn a>ut-h|.
\end{otherlanguage*}
```

Altro testo in italiano `\dots`

... testo in italiano.

Τη πάντα διδούση καὶ ἀπολαμβάνουσα φύσει ὁ πεπαιδευμένος καὶ αἰδήμων λέγει· “ὄδς, ὃ θέλεις, ἀπόλαβε, ὃ θέλεις”. Λέγει δὲ τοῦτο οὐ καταθρασυνόμενος, ἀλλὰ πειθαρχῶν μόνον καὶ εὐνοῶν αὐτῆ.

Altro testo in italiano...

e si può inserire un testo breve o brevissimo in un capoverso scrivendolo nell'argomento di un comando personale `\greco` (leggermente diverso dall'analogo `\latino`):

```
\newcommand{\greco}{\foreignlanguage{greek}}
```

da usare come segue:

```
In lettere minuscole, \TeX{}
si scriverebbe \greco{teq}.
È la radice della parola greca
\greco{t'eqnh} che vuol dire
''arte'' e ''mestiere''.
```

In lettere minuscole, \TeX si scriverebbe $\tau\epsilon\chi$. È la radice della parola greca $\tau\acute{\epsilon}\chi\eta$ che vuol dire “arte” e “mestiere”.

B.2.1 Traslitterare i caratteri

Greco antico

Come si può osservare negli esempi precedenti, il sorgente “in greco” è un insieme di caratteri latini e altri segni (tutti conosciuti, ma qui con funzioni diverse) che poco assomiglia a ciò che si vede a destra. Lo si è scritto, infatti, con il metodo più immediato per chi possiede una tastiera italiana, cioè la *traslitterazione dei caratteri*. La figura 17 mostra (in blu) come l'opzione `greek` di `babel` interpreta i caratteri latini (alfabetici e non) che hanno una corrispondenza in greco. Tutti gli altri, compresi i caratteri speciali di \LaTeX , sono uguali nelle due lingue (per la tilde, però, si veda sotto). Si noti che la lettera italiana *v* non ha un corrispettivo greco perché è una specie di carattere greco “speciale”: la scrittura *sv*, infatti, è l'unico modo per ottenere

Tabella 58: Diacritici e punteggiatura del greco antico.

Segno		Codice	Esempio	Risultato
Accento	acuto	'	'w	ώ
	grave	`	`w	ὠ
	circonflesso	~	~w	ῶ
Spirito	dolce	>	>w	ὠ
	aspro	<	<w	ῶ
Coronide		>	t>~alla	τῶλλα
Dieresi		"	"i	ῖ
ι sottoscritto			w	Ϝ
Apostrofo		"_	d" >eg'w	δ' ἐγώ
Punto	in alto	;		·
	e virgola	?		;

un sigma iniziale o mediale isolato σ , altrimenti interpretato dal programma sempre come finale ς .

La tabella 58 raccoglie i segni di punteggiatura e i diacritici necessari per scrivere in greco. Si noti che:

- I font greci predefiniti permettono di combinare i diacritici in *tutti* i modi previsti dalla grammatica semplicemente premettendoli a una lettera *in una sequenza qualsiasi*.
- L'unico diacritico da posporre alla lettera è lo ι sottoscritto.
- Il carattere " nel sorgente restituisce un apostrofo soltanto se seguito da uno spazio (mantenuto anche nel testo composto, a differenza dell'italiano): "_.
- Il carattere ~, che in greco produce l'accento circonflesso, potrebbe *talvolta* produrre il consueto spazio indivisibile (che scrivendo in greco si ottiene con `\nobreakspace`), nelle seguenti sequenze: ~, ~", "~", ~<, <~, ~> e >~. Si risolve il problema premettendo a ciascuna una barra rovescia e caricando nel preambolo il file `\graccents-glyphs.def` con il comando `\input`.
- Il greco antico non prevedeva nella scrittura apici e virgolette, che si possono comunque ottenere con i codici ‘ ‘ ’ per gli apici, (() per le virgolette caporali e << >> per quelle inglesi.

L'opzione `greek` di `babel` definisce ulteriori caratteri utili per scopi particolari:

- i caratteri minuscoli per indicare alcuni numeri (il *qoppa* ϱ ottenuto con `\qoppa` per il numero 6, lo *stigma* ς con `\stigma` per il 90 e il *sampi* $\var�$ con `\sampi` per il 900); quelli per il *digamma* minuscolo $\var�$, ottenuto con `\ddigamma` e maiuscolo Φ , ottenuto con `\Digamma`; il carattere ξ , ottenuto con `\vardigamma`;
- l'apice ' e il pedice , per scrivere i numerali, ottenuti rispettivamente con `\anwtonos` e `\katwtonos`;
- i due comandi `\greeknumeral` e `\Greeknatural`, che trasformano automaticamente il numero arabo scritto nel loro argomento (da 1 a 999 999, in greco lo zero non esisteva) nel corrispondente numero greco secondo il sistema di numerazione alfabetico, minuscolo e maiuscolo rispettivamente (ma si noti che *qoppa*, *stigma* e *sampi* saranno *sempre* minuscoli):

```

\begin{otherlanguage*}{greek}
\greeknumeral{996} \ \
\Greeknuneral{123}
\end{otherlanguage*}

```

λητ'
 PKΓ'

Greco moderno

Dai primi anni Ottanta del secolo scorso, in Grecia il sistema politonico ha lasciato il posto a quello monotonico, che nella varietà scritta presenta un solo accento (dipende dalle scelte tipografiche degli editori, ma di solito è quello acuto).

L'ortografia monotonica è molto più semplice dell'altra, avendo bisogno di soli altri due segni oltre a lettere e punteggiatura:

- ' per l'accento acuto;
- " (lo spazio, evidenziato, è necessario) per l'apostrofo, oppure " seguito da ι/I o υ/Υ per la dieresi.

Si dichiara nel preambolo caricando babel *senza alcun attributo*:

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

e un testo in questa varietà di greco appare così:

```

\begin{otherlanguage*}{greek}
Pr'epei na skefto'ume mia
sun'arthsh ths opo'ias
gnwr'izoume 'oti up'arqei
to olokl'hrwma.
\end{otherlanguage*}

```

Πρέπει να σκεφτούμε μια συνάρτηση
 της οποίας γνωρίζουμε ότι υπάρχει το
 ολοκλήρωμα.

Si noti che nonostante greek preveda tre distinti schemi di sillabazione per il greco (due per il greco politonico, antico e moderno, e uno per il greco monotonico), di fatto babel usa *sempre* quello per il greco politonico antico.

B.2.2 Scrivere in greco nel sorgente

Il modo appena descritto per scrivere in greco non è particolarmente difficile: la corrispondenza di lettere e diacritici sulla tastiera si memorizza in poco tempo, ma la traslitterazione rende comunque il sorgente difficile da leggere, specie per chi il greco lo conosce.

Quando in un documento le parti in greco sono lunghe o lunghissime (o quando il documento è in greco per intero) poterle scrivere direttamente in lingua diventa una comodità quasi irrinunciabile. I metodi per farlo sono più d'uno: si può usare una tastiera bilingue o una tastiera virtuale che non richieda di ricordare tutti gli abbinamenti, oppure si può ridefinire allo scopo i tasti di una tastiera italiana (con i modi propri di ogni sistema operativo). Qualunque metodo si scelga tra quelli spiegati in questo paragrafo, l'editor in uso deve supportare *pienamente* la codifica Unicode.

Per comporre il documento *con* \LaTeX usando i font greci predefiniti, a T1 va aggiunta la codifica LGRx. Scritto in greco, l'esempio iniziale diventa:

```

\documentclass[a4paper]{article}
\usepackage[LGRx,T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[plutonikogreek,italian]{babel}

\begin{document}

```

```

\begin{otherlanguage*}{greek}
Τη πάντα διδούση καὶ ἀπολαμβανούση φύσει ὁ πεπαιδευμένος καὶ αἰδήμων
λέγει· “δὸς, ὁ θέλεις, ἀπόλαβε, ὁ θέλεις”. Λέγει δὲ τοῦτο οὐ
καταθρασυνόμενος, ἀλλὰ πειθαρχῶν μόνον καὶ εὐνοῶν αὐτῆ.
\end{otherlanguage*}

\end{document}

```

L’alternativa è il programma Xe_{La}T_EX (da scegliere tra i motori gestiti dall’editor in uso), con il quale è facile usare i font già presenti nel proprio sistema operativo, che richiede un preambolo come il seguente:

```

\documentclass[a4paper]{article}
\usepackage{fontspec}
\setmainfont{Linux Libertine O}
\usepackage{polyglossia}
\setmainlanguage{italian}
\setotherlanguage[variant=ancient]{greek}

\begin{document}

\begin{greek}
Τη πάντα διδούση καὶ ἀπολαμβανούση φύσει ὁ πεπαιδευμένος καὶ αἰδήμων
λέγει· “δὸς, ὁ θέλεις, ἀπόλαβε, ὁ θέλεις”. Λέγει δὲ τοῦτο οὐ
καταθρασυνόμενος, ἀλλὰ πειθαρχῶν μόνον καὶ εὐνοῶν αὐτῆ.
\end{greek}

\end{document}

```

Si noti che:

- il pacchetto fontenc è sostituito da fontspec, che gestisce i font del sistema in modo molto potente;
- il pacchetto inputenc *non* va caricato, perché Xe_{La}T_EX lavora con la codifica UTF-8 per impostazione predefinita;
- `\setmainfont` imposta il font principale del documento con le eventuali opzioni (qui si è scelto il font Linux Libertine, già presente nel sistema e dotato di una collezione completa di caratteri greci);
- il pacchetto babel è sostituito da polyglossia;
- `\setmainlanguage` imposta la lingua principale del documento con le eventuali opzioni;
- `\setotherlanguage` carica l’altra lingua del documento con le eventuali opzioni, in questo caso il greco antico nella variante *ancient*, cioè politonico (si possono caricare in una volta sola più lingue *senza opzioni* con il comando `\setotherlanguages`);
- l’ambiente `greek` equivale a `otherlanguage*{greek}`.

A differenza di babel, polyglossia carica *effettivamente* la sillabazione corrispondente alla variante indicata: *ancient* (o *poly*) per il greco politonico, *mono* per quello monotonico.

Si noti che usando Xe_{La}T_EX nelle parti in italiano microtype permette solo la protrusione dei caratteri, ma non l’espansione e altre finzze.

B.2.3 Documenti interamente in greco

Chi dovesse scrivere un documento completamente in greco (seguendo indifferentemente l'una o l'altra ortografia) può contare su qualche agevolazione in più. L'opzione `greek` di `babel`, infatti:

- definisce due comandi per convertire automaticamente la data corrente nella corrispondente data greca: `\dategreek` la mette in numerali arabi (23 Νοεμβρίου 2012), `\Grtoday` in numerali alfabetici (ΚΤ' Νοεμβρίου ,ΒΙΒ');
- traduce in greco le parole fisse delle classi di documento standard in accordo con l'ortografia dichiarata: *Indice* diventa Εύρετήριο in greco politonico, ma Ευρετήριο in greco monotonico, per esempio;
- definisce altri due comandi utili: `\euro`, che produce il simbolo dell'euro €, e `\permill`, che produce il simbolo di per mille ‰.

Si fornisce infine il preambolo tipo per comporre un documento in italiano e greco o nel solo greco con \LaTeX (le righe sono numerate per maggiore chiarezza).

```

1 \documentclass[a4paper]{article}
2 \usepackage[LGRx,T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \input{lgraccents-glyphs.def}
5
6 \usepackage[(polutoniko)greek,italian]{babel}
7 \usepackage[(polutoniko)greek]{babel}
8
9 \begin{document}
10
11 ...
12
13 \end{document}

```

Si noti che:

- va indicata obbligatoriamente *una sola* tra le righe 6 e 7 a seconda delle o della lingue del documento;
- si metta `polutonikogreek` anziché `greek` soltanto per avere la corretta traduzione delle parole fisse e la corretta interpretazione di ~.

Naturalmente, anche per \XeLaTeX valgono le osservazioni di questo paragrafo: basterà modificare il preambolo esemplificato nel paragrafo B.2.2 a pagina 204 impostando il greco come lingua principale del documento con

```
\setmainlanguage[opzioni]{greek}
```

B.2.4 Specialità: teubner

Chi avesse la necessità di comporre documenti in greco di altro tipo (in particolare linguistici, filologici e letterari), può usare il pacchetto `teubner`, che comprende una collezione di caratteri inclinati particolarmente raffinati, disegnati originariamente nella tipografia Teubner di Lipsia (in onore della quale è stato scelto il nome del pacchetto), e tra le altre cose permette di regolare molto finemente le combinazioni dei diacritici sulle lettere e le legature tra i caratteri.

Si raccomanda infine di caricare `teubner` *dopo* `amsmath`, se dovesse servire anche quest'ultimo pacchetto. Per maggiori dettagli si rimanda alla sua documentazione.



C.1 INTRODUZIONE

Nelle prossime pagine si descriverà nei suoi elementi essenziali beamer, una classe di documento molto potente e altamente versatile dedicata alle videopresentazioni. Fra i suoi punti di forza ci sono la gestione semplice degli effetti dinamici tra le diapositive, la possibilità di sezionare il testo con i comandi standard, la piena compatibilità con gli altri pacchetti e un ventaglio di personalizzazioni virtualmente illimitato. Si rimanda chi voglia approfondire l'argomento alla corposa e dettagliata documentazione della classe e a [Pantieri e Gordini, 2009], da cui queste note sono tratte.

Si noti che beamer carica automaticamente i pacchetti hyperref, graphicx, amsmath, amssymb, amsthm.

Qualche consiglio

Una presentazione efficace è il risultato dell'esposizione chiara di contenuti essenziali, possibilmente riprodotti con diapositive poco elaborate: beamer aiuta, ma da solo non basta. Si tenga presente di:

- scrivere e parlare il più semplicemente possibile;
- ridurre il contenuto della diapositiva al minimo indispensabile (il pubblico deve ascoltare chi parla e non leggere);
- proiettare una diapositiva al minuto, non di più;
- suddividere la presentazione in frazioni di circa 5-7 minuti l'una;
- usare un font senza grazie e di corpo molto grande (leggibile senza fatica anche nelle ultime file);
- evitare capoversi giustificati (difficili da ottenere e inutili);
- usare lo stesso formato per tutte le diapositive.

C.2 FONDAMENTALI

In linea generale, una presentazione realizzata con beamer consiste in una serie di *quadri* (*frame*, in inglese), ciascuno dei quali è composto da una o più *diapositive* (*slide*).

c.2.1 Per cominciare

L'esempio

```
\documentclass{beamer}

% dati generali
\title{La nostra prima presentazione}
\author{Lorenzo Pantieri \and Tommaso Gordini}
\date{1 maggio 2011}
```

```

\begin{document}

% quadro 1
\begin{frame}
\maketitle
\end{frame}

% quadro 2
\begin{frame}
\frametitle{Un esempio}
\begin{itemize}
\item<1-> Mane
\item<2-> Tekel
\item<3-> Fares
\end{itemize}
\end{frame}

\end{document}

```

riproduce il codice della presentazione di quattro diapositive in due quadri mostrata nella figura 18 nella pagina successiva: il primo quadro ne contiene il titolo generale; il secondo un elenco di tre elementi. Si osservi che le cose ancora sconosciute sono poche:

- l'ambiente `frame` racchiude un quadro;
- il comando `\frametitle` produce il titolo del quadro: *ogni* quadro dovrebbe averne uno;
- se le diapositive di un quadro sono più d'una, si mettono in un ambiente `itemize`, dove ogni `\item` produce una diapositiva;
- le espressioni racchiuse tra i segni di `<` e `>` verranno spiegate nel paragrafo C.2.3 a pagina 212.

Anche in questo caso, infine, un uso oculato dei commenti facilita la lettura del sorgente.

c.2.2 Una presentazione articolata

Il professor Euclide dell'Università di Alessandria d'Egitto è pronto per rivelare al mondo la sua ultima scoperta: i numeri primi sono infiniti. Euclide parlerà durante il VII Simposio Internazionale sui Numeri Primi dove avrà a disposizione venti minuti, cinque dei quali riservati alle domande del pubblico. La presentazione che ha preparato è la seguente:

```

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

% dati generali
\title{I numeri primi sono infiniti}
\author[Euclide]{Euclide di Alessandria \\\
\texttt{euclide@alessandria.edu}}
\date[VII SINP]{VII Simposio Internazionale sui Numeri Primi}
\institute[UniAlessandria]{Università di Alessandria}
\logo{\includegraphics[width=0.2\textwidth]{sigillo}}

% temi e aspetto del testo

```

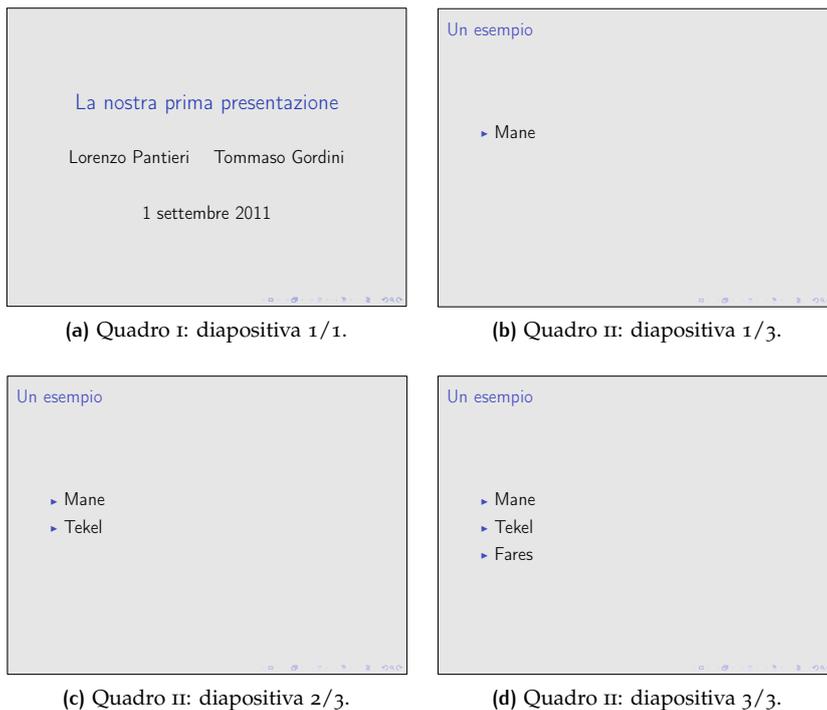


Figura 18: Presentazione semplice.

```

\usetheme{AnnArbor}
\useoutertheme[right]{sidebar}
\setbeamercovered{dynamic}

% definizione degli enunciati matematici
\theoremstyle{definition}
\newtheorem{definizione}{Definizione}
\theoremstyle{plain}
\newtheorem{teorema}{Teorema}

\begin{document}

% quadro 1
\begin{frame}
\maketitle
\end{frame}

% quadro 2
\begin{frame}
\frametitle{Piano della presentazione}
\tableofcontents
\end{frame}

% quadro 3
\section{Introduzione}
\begin{frame}
\frametitle{Che cosa sono i numeri primi?}
\begin{definizione}
Un \alert{numero primo} è un intero  $>1$  che ha esattamente
due divisori positivi.
\end{definizione}
\end{frame}

```

```

% quadro 4
\section{L'infinità dei primi}
\begin{frame}
\frametitle{I numeri primi sono infiniti}
\framesubtitle{Ne diamo una dimostrazione diretta}
\begin{teorema}
Non esiste un primo maggiore di tutti gli altri.
\end{teorema}
\pause
\begin{proof}
\begin{enumerate}[<+>]
\item Sia dato un elenco di primi.
\item Sia  $q$  il loro prodotto.
\item Allora  $q+1$  è divisibile per un primo  $p$ 
che non compare nell'elenco. \qedhere
\end{enumerate}
\end{proof}
\end{frame}

% quadro 5
\section{Problemi aperti}
\begin{frame}
\frametitle{Che cosa c'è ancora da fare?}
\begin{block}{Problemi risolti}
Quanti sono i numeri primi?
\end{block}
\begin{block}{Problemi aperti}
Un numero pari  $>2$  è sempre la somma di due primi?
\end{block}
\end{frame}

\end{document}

```

è una presentazione di otto diapositive in cinque quadri (mostrata nella figura 19 nella pagina successiva) che di seguito si analizza negli elementi nuovi.

Preambolo

Contiene come al solito le istruzioni generali del documento. Si noti che:

- gli argomenti facoltativi di `\author` e `\date` mettono il loro contenuto in punti particolari della diapositiva (di solito in basso).
- `\institute` (il cui argomento facoltativo si comporta come descritto sopra) inserisce nella diapositiva il nome dell'istituzione di afferenza e `\logo` ne mette il logo. (Se lo si desidera, si può anche inserire una figura nella diapositiva iniziale scrivendo nel preambolo l'istruzione `\titlegraphic{\includegraphics{\figura}}`.)
- Con `\usetheme` si sceglie il *tema* della presentazione, cioè l'aspetto generale delle diapositive per colori e disposizione degli elementi, e con `\useoutertheme` si scelgono ulteriori elementi di contorno (si veda il paragrafo C.3 a pagina 217).
- `\setbeamercovered` regola l'aspetto del testo nella proiezione. Accetta tre valori: con `transparent` il testo non ancora proiettato è (semi)trasparente (quel tanto che basta a suggerire a chi parla come proseguire l'esposizione); con `invisible` (predefinito) è completamente nascosto; con `dynamic` è tanto più trasparente quanto più tempo deve rimanere nascosto, e viceversa.

I numeri primi sono infiniti

Euclide di Alessandria
euclide@alessandria.edu
Università di Alessandria

VII Simposio Internazionale sui Numeri Primi

(a) Quadro I: diapositiva 1/1.

I numeri primi sono infiniti
Ne diamo una dimostrazione diretta

Teorema
Non esiste un primo maggiore di tutti gli altri.

Dimostrazione.

- 1 Sia dato un elenco di primi.
- 2 Sia q il loro prodotto.
- 3 Allora $q + 1$ è divisibile per un primo p che non compare nell'elenco.

(e) Quadro IV: diapositiva 2/4.

Piano della presentazione

- 1 Introduzione
- 2 L'infinità dei primi
- 3 Problemi aperti

(b) Quadro II: diapositiva 1/1.

I numeri primi sono infiniti
Ne diamo una dimostrazione diretta

Teorema
Non esiste un primo maggiore di tutti gli altri.

Dimostrazione.

- 1 Sia dato un elenco di primi.
- 2 Sia q il loro prodotto.
- 3 Allora $q + 1$ è divisibile per un primo p che non compare nell'elenco.

(f) Quadro IV: diapositiva 3/4.

Che cosa sono i numeri primi?

Definizione
Un numero primo è un intero > 1 che ha esattamente due divisori positivi.

(c) Quadro III: diapositiva 1/1.

I numeri primi sono infiniti
Ne diamo una dimostrazione diretta

Teorema
Non esiste un primo maggiore di tutti gli altri.

Dimostrazione.

- 1 Sia dato un elenco di primi.
- 2 Sia q il loro prodotto.
- 3 Allora $q + 1$ è divisibile per un primo p che non compare nell'elenco.

(g) Quadro IV: diapositiva 4/4.

I numeri primi sono infiniti
Ne diamo una dimostrazione diretta

Teorema
Non esiste un primo maggiore di tutti gli altri.

Dimostrazione.

- 1 Sia dato un elenco di primi.
- 2 Sia q il loro prodotto.
- 3 Allora $q + 1$ è divisibile per un primo p che non compare nell'elenco.

(d) Quadro IV: diapositiva 1/4.

Che cosa c'è ancora da fare?

Problemi risolti
Quanti sono i numeri primi?

Problemi aperti
Un numero pari > 2 è sempre la somma di due primi?

(h) Quadro V: diapositiva 1/1.

Figura 19: Presentazione complessa.

Primo quadro

Composto di una sola diapositiva (figura 19a), contiene il titolo della presentazione. In basso a destra si vede il contatore dei quadri in forma di frazione (1/5): quello proiettato, dunque, è il primo di cinque quadri in totale.

Secondo quadro

Composto di una sola diapositiva (figura 19b), contiene l'indice generale dell'esposizione, utile all'inizio per illustrare il piano della proiezione. L'indice è riportato esattamente sulla *barra di navigazione* laterale, le cui voci (cliccabili per portarsi rapidamente da un punto all'altro della presentazione) s'illuminano progressivamente all'esaurirsi dei quadri.

Intermezzo

Tra il secondo e il terzo quadro c'è un comando di sezionamento standard che va dato, si noti bene, *prima* che il quadro cominci. In beamer questi comandi servono solo a comporre l'indice generale: il contenuto del loro argomento non verrà *mai* mostrato. Si ricordi che una presentazione non è un documento tradizionale, per cui si raccomanda di valutarne attentamente la struttura e di non sezionarla eccessivamente.

Terzo quadro

Composto di una sola diapositiva (figura 19c), contiene una definizione in cui due parole sono evidenziate con il comando `\alert`, che marca il proprio argomento diversamente dai comandi standard (in blu, in questo caso).

Quarto quadro

È il quadro più articolato, composto di quattro diapositive (figure 19d-g). Si noti che qualunque sia il numero di diapositive che lo compongono, il suo contenuto va scritto *una sola volta*: le tecniche di esposizione incrementale spiegate nel prossimo paragrafo si occuperanno di dilazionarne la proiezione.

Quinto quadro

Composto di una sola diapositiva, evidenzia il proprio contenuto in due *blocchi* (i due ambienti `block`), per i quali si rimanda al paragrafo C.2.4 a pagina 214.

c.2.3 Esposizione incrementale

Rivelare progressivamente il contenuto di un quadro (specie se si tratta di un ragionamento svolto per punti o di un elenco) proiettandone gli elementi uno alla volta può aiutare il pubblico a seguire meglio il discorso e a limitare il carico d'attenzione. Con beamer questo effetto si ottiene in diversi modi, anche se si raccomanda di non abusarne e di riservarlo ai casi di vera necessità.

Il metodo più semplice è frazionare il contenuto del quadro dando nei punti opportuni altrettanti comandi `\pause`. Ogni diapositiva mostrerà il testo corrente più tutto quello precedente, e così via.

Il secondo metodo è dare dopo ogni `\item` il comando `\onslide`, la cui sintassi generale è

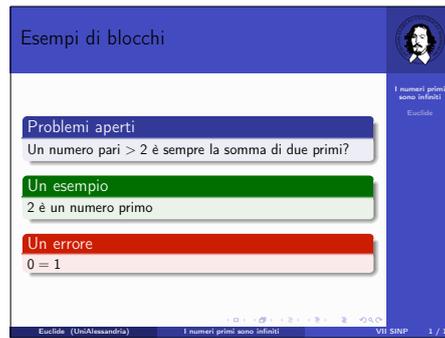


Figura 20: Esempi di blocchi.

```
\onslide<intervallo>{<testo>}
```

dove il *<testo>* verrà mostrato solo nelle diapositive indicate nell'*<intervallo>*. Nelle altre, anche se non presente sullo schermo, avrà comunque lo spazio corrispondente. Per esempio, un elenco i cui punti scompaiano uno dopo l'altro dopo essere stati proiettati si ottiene con:

```
\begin{frame}
\begin{itemize}
\item \onslide<1>{Mane}
\item \onslide<2>{Tekel}
\item \onslide<3>{Fares}
\end{itemize}
\end{frame}
```

Il terzo metodo consiste nello scrivere subito dopo ciascun `\item` un'espressione che nella sua forma più completa è:

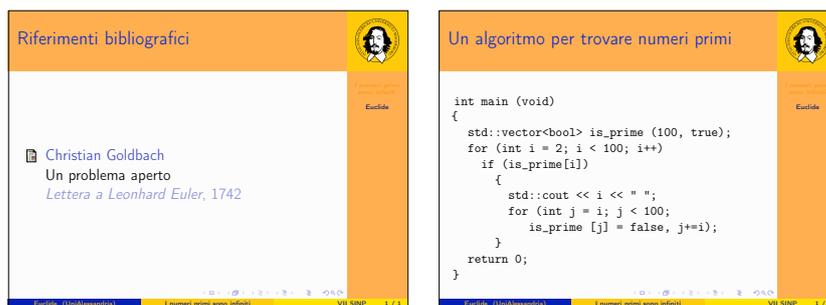
```
<numero>-<numero>
```

dove:

- `<numero>` è il numero progressivo della diapositiva nella presentazione;
- un semplice numero (`<4>`, per esempio) fa proiettare l'elemento una sola volta nella quarta diapositiva del quadro;
- un intervallo numerico separato con un trattino (`<4-6>`, per esempio) mantiene l'elemento visibile nelle diapositive dalla quarta alla sesta comprese, ma non prima né dopo;
- non scrivere nulla prima del trattino (`<-4>`) equivale a scrivere `<1-4>`;
- non scrivere nulla dopo il trattino (`<4->`) rende visibile l'elemento dalla diapositiva in cui compare la prima volta (la quarta, in questo caso) fino all'ultima diapositiva del quadro (come nella prima presentazione analizzata).

Si poteva scrivere più sinteticamente il secondo quadro della prima presentazione con:

```
\begin{itemize}[<->]
\item Mane
\item Tekel
\item Fares
\end{itemize}
```



(a) Diapositiva con riferimento bibliografico.

(b) Diapositiva con codice.

Figura 21: Diapositive particolari.

c.2.4 Blocchi

In beamer un *blocco* è una cornice più o meno marcata che evidenzia una porzione di testo. La classe definisce tre ambienti standard per i blocchi, resi diversamente in base al tema caricato: `block` per i blocchi generici, `exampleblock` per gli esempi e `alertblock` per gli avvisi. Per esempio, in una presentazione con il tema Madrid il codice

```
\begin{frame}
\begin{block}{Problemi aperti}
Un numero pari  $>2$  è sempre la somma di due primi?
\end{block}

\begin{exampleblock}{Un esempio}
 $2$  è un numero primo
\end{exampleblock}

\begin{alertblock}{Un errore}
 $0=1$ 
\end{alertblock}
\end{frame}
```

produce la diapositiva mostrata nella figura 20 nella pagina precedente, in cui si vede che la differenza fra i blocchi è soltanto cromatica. Per le innumerevoli personalizzazioni possibili si veda la documentazione della classe.

c.2.5 Bibliografia

Non dovrebbe *mai* essere necessario proiettare i riferimenti bibliografici del proprio lavoro. Nei rari casi in cui servisse, si può usare un codice come

```
\begin{frame}
\frametitle{\refname}
\begin{thebibliography}{9}
\bibitem{goldbach:congettura} Christian Goldbach
\newblock Un problema aperto
\newblock \emph{Lettera a Leonhard Euler}, 1742
\end{thebibliography}
\end{frame}
```

che produce la diapositiva mostrata nella figura 21a, dove il comando standard `\newblock` separa i diversi elementi della bibliografia. Va da sé che in una presentazione non vanno *mai* usati nemmeno i comandi come `\cite`

o analoghi, perché durante la proiezione non avrebbe modo di verificare a quale opera si riferiscono.

c.2.6 Codici

Per scrivere del codice in una diapositiva, come mostra la figura 21b a fronte, si può usare l'ambiente standard `verbatim`, ricordandosi di dare a `frame` l'opzione `fragile`:

```

\begin{frame}[fragile]
\frametitle{Un algoritmo per trovare numeri primi}
\begin{verbatim}
int main (void)
{
    std::vector<bool> is_prime (100, true);
    for (int i = 2; i < 100; i++)
        if (is_prime[i])
            {
                std::cout << i << " ";
                for (int j = i; j < 100; is_prime [j] = false, j+=i);
            }
    return 0;
}
\end{verbatim}
\end{frame}

```

c.2.7 Disporre il contenuto su più colonne

Disporre il contenuto di una diapositiva su più colonne è utile soprattutto quando si devono inserire tabelle o figure con la relativa descrizione, che in una presentazione va messa sempre *accanto*. Gli ambienti mobili `table` e `figure`, infatti, non hanno motivo di essere usati: chi parla non dirà mai «come abbiamo visto nella figura n», perché *nessuno* durante la spiegazione si ricorderà qual è la figura n.

Le diverse colonne sono prodotte da altrettanti ambienti `column` (ai quali va *sempre* assegnata una larghezza) annidati in un unico ambiente `columns`. Il codice seguente

```

\begin{frame}
\begin{columns}
\begin{column}{0.4\textwidth}
Mane \\ Tekel \\ Fares
\end{column}

\begin{column}{0.4\textwidth}
Una riga (centrata)
\end{column}
\end{columns}
\end{frame}

```

produce la diapositiva mostrata nella figura 22a nella pagina successiva, nella quale le righe delle due colonne sono centrate verticalmente l'una rispetto all'altra (è l'impostazione predefinita). Se si vogliono le prime righe di ciascuna colonna allineate come nella figura 22b nella pagina seguente, invece, basta dare a `columns` l'opzione `t`.

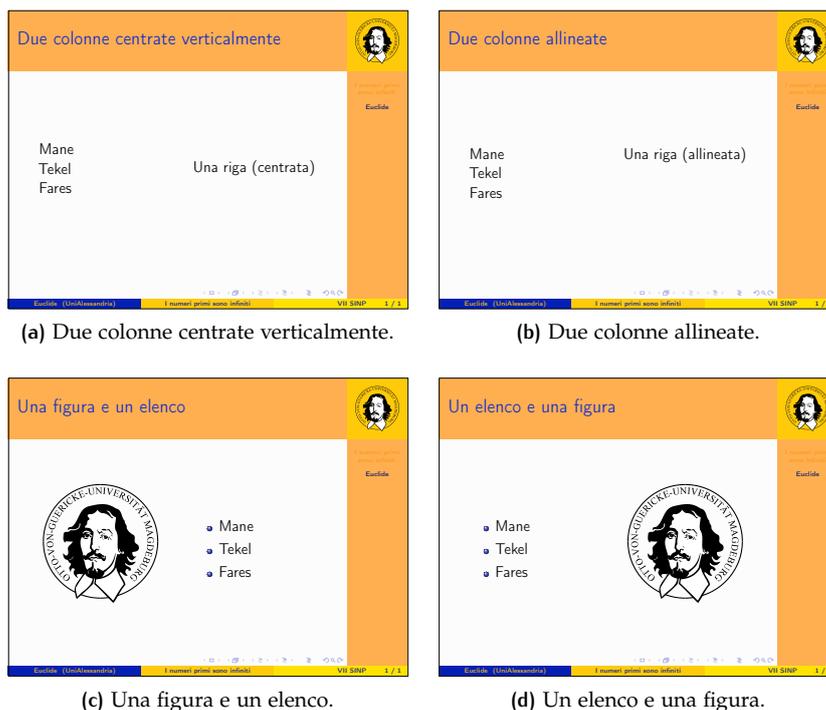


Figura 22: Diapositive strutturate su più colonne.

Una figura (a sinistra) con accanto un elenco (a destra), come nella figura 22c, si ottiene con

```
\begin{frame}
\begin{columns}
\begin{column}{0.4\textwidth}
\includegraphics[width=\columnwidth]{figura}
\end{column}

\begin{column}{0.4\textwidth}
\begin{itemize}
\item Mane
\item Tekel
\item Fares
\end{itemize}
\end{column}
\end{columns}
\end{frame}
```

Si ottiene l'effetto contrario (mostrato nella figura 22d) invertendo il contenuto dei due ambienti column.

c.2.8 Stampare la presentazione

Assegnando a beamer l'opzione `handout` si ottiene facilmente anche una versione della presentazione adatta alla stampa (un *handout*, in gergo), ovviamente priva di (eventuali) effetti dinamici. A questo scopo il pacchetto `pgfpages` permette di raccogliere in una stessa pagina 2, 4, 8 o 16 diapositive caricandolo così:

```
\usepackage{pgfpages}
\pgfpagesuselayout{4 on 1}[a4paper,border shrink=5mm,landscape]
```



Figura 23: Alcuni temi predefiniti di beamer.

dove si chiede a \LaTeX di mettere quattro diapositive in ogni pagina (4 on 1) di formato A4 disposta in orizzontale (landscape), con un piccolo spazio (border shrink) di 5 mm attorno a ciascuna diapositiva. (L'opzione landscape è richiesta soltanto se in una pagina si vogliono 4 o 16 diapositive, non negli altri casi.)

C.3 PRESENTAZIONI PERSONALIZZATE

In questa sezione si mostreranno solo alcuni esempi delle numerose possibilità offerte da beamer per personalizzare le presentazioni.

Il *tema* gestisce l'aspetto grafico della presentazione definendo i colori che appariranno nelle diapositive, il formato di elenchi e blocchi e infine l'aspetto e la disposizione degli elementi di contorno. Si noti che tutti questi elementi possono essere regolati uno per uno, moltiplicando le possibili personalizzazioni. Se non dovessero bastare i temi precaricati in beamer, se ne possono scaricare moltissimi altri da Internet o ancora crearsene di personali da zero.

Il tema si seleziona scrivendo nel preambolo

```
\usetheme{<tema della presentazione>}
```

dove *<tema della presentazione>* è il nome del tema prescelto che, per cominciare, si può scegliere tra quelli predefiniti dalla classe, identificati di regola dal nome di una città.

Non potendoli qui riportare tutti, si tenga presente che a grandi linee i temi possono essere:

- senza barra di navigazione (AnnArbor, CambridgeUS, Madrid, per esempio);

- con barra di navigazione laterale (Berkeley, Goettingen, Marburg);
- con barra di navigazione ad albero (Antibes, JuanLesPins, Montpellier);
- con quadro di navigazione (Berlin, Dresden, Warsaw, Singapore).

Alcuni di essi vengono mostrati nelle diapositive della figura [23](#) nella pagina precedente.

ACRONIMI

- AMS** American Mathematical Society
Fondata nel 1888, con svariate decine di migliaia di soci la Società Matematica Americana è una delle più importanti associazioni di matematici nel mondo. L'AMS ha sostenuto attivamente lo sviluppo di L^AT_EX ed è stata tra i primi organismi scientifici a sollecitare gli autori a scrivere con questo programma.
- CTAN** Comprehensive T_EX Archive Network
La "rete di archivi completi di T_EX" è, nel Web, il luogo di riferimento da cui scaricare software e materiale su T_EX e L^AT_EX.
- G.IT** Gruppo Utilizzatori Italiani di T_EX e L^AT_EX
È un'associazione senza fini di lucro con lo scopo di aumentare la diffusione di T_EX e L^AT_EX in Italia attraverso la condivisione di informazioni legate al loro uso, conciliando il vantaggio dell'apprendimento con il piacere dell'insegnamento.
- HTML** Hyper Text Mark-up Language
Il "linguaggio di marcatura degli ipertesti" è un linguaggio di pubblico dominio inventato da Tim Berners-Lee al CERN di Ginevra nel 1989 per descrivere i documenti ipertestuali che popolano il Web.
- ISO** International Standard Organization
L'"organizzazione internazionale per le standardizzazioni" è la più importante associazione a livello mondiale per la definizione di standard tecnico-scientifici. Suoi membri sono gli organismi nazionali di standardizzazione di 162 Paesi del mondo.
- JPEG** Joint Photographic Experts Group
È il più usato standard di compressione delle immagini fotografiche. Eseguisce una compressione con perdita di informazioni (cioè di tipo *lossy*). Insieme con il PNG, è il formato standard delle immagini bitmap da inserire in un documento da comporre con L^AT_EX.
- PDF** Portable Document Format
È il formato di file più versatile per la stampa e la distribuzione elettronica, introdotto dalla Adobe Systems nel 1993 per rappresentare documenti indipendentemente dall'hardware e dal software usati per generarli o visualizzarli. Il PDF eredita molte delle funzioni del PostScript, un linguaggio di descrizione della pagina sviluppato dalla stessa azienda. È il formato standard delle immagini vettoriali da inserire in un documento da comporre con L^AT_EX.
- PNG** Portable Network Graphics
Creato nel 1995, è un formato di file particolarmente adatto per rappresentare disegni e icone. Eseguisce una compressione senza perdita di informazioni (cioè di tipo *lossless*). Con il JPG, è il formato standard delle immagini bitmap da inserire in un documento da comporre con L^AT_EX.
- TUG** T_EX User Group
L'espressione ("gruppo di utenti di T_EX") indica le associazioni di persone accomunate dalla passione per T_EX e L^AT_EX sparse nei principali Paesi del mondo. Il loro scopo è quello di diffondere l'uso di questi due programmi e di fornire supporto alle rispettive comunità di utenti.

- UNI** Ente Nazionale Italiano di Unificazione
È un'associazione privata senza scopo di lucro che svolge attività normativa in tutti i principali settori tecnico-scientifici. Rappresenta l'Italia in seno all'ISO.
- URL** Uniform Resource Locator
È una stringa di caratteri che identifica in modo univoco l'indirizzo di una qualunque risorsa in Rete. Ogni URL (per esempio <http://www.tug.org/texlive/>) si compone normalmente di tre parti: il protocollo usato per indirizzare la risorsa (http, nell'esempio considerato), il nome dell'*host* o del server o del dominio (www.tug.org), e infine il nome del file che contiene la risorsa (/texlive/).
- UTF** Unicode Transformation Format
Unicode è un sistema di codifica che assegna una combinazione di bit a ogni carattere indipendentemente da programma, piattaforma e lingua usati. La codifica UTF-8 (*Unicode Transformation Format-8 bit*) è una particolare realizzazione di Unicode.
- WYSIWYG** What You See Is What You Get
L'espressione indicata dall'acronimo ("ciò che vedi è ciò che ottieni") ha sostanzialmente due significati.
Il primo si riferisce al problema di ottenere in stampa testo e immagini che abbiano una disposizione grafica uguale a quella visualizzata sullo schermo del calcolatore. I primi software e le prime stampanti per uso domestico non davano risultati pienamente soddisfacenti e si superò il problema introducendo nuovi dispositivi e software (pionieri furono il sistema di codifica dei caratteri TrueType sviluppato dalla Apple e il programma Adobe TypeManager).
Con il tempo, il significato dell'acronimo si è esteso per analogia anche ad alcune problematiche connesse alla creazione dei documenti. Nei comuni elaboratori di testo (come per esempio Microsoft Word), l'utente agisce direttamente sul testo già composto così come appare sullo schermo, e ogni sua azione si traduce in un'immediata variazione di quel testo. Programmi di questo tipo vengono detti WYSIWYG (in questo secondo significato), e il tipo di composizione che adottano viene denominato "composizione sincrona". L'acronimo che si riferisce al concetto opposto è WYSIWYM.
- WYSIWYM** What You See Is What You Mean
L'acronimo ("ciò che vedi è ciò che intendi") è stato coniato espressamente per L^AT_EX in contrapposizione a WYSIWYG (nel senso di "programma di videoscrittura caratterizzato da una composizione sincrona").
La caratteristica che più differenzia L^AT_EX dagli altri elaboratori di testo è il fatto che per realizzare un documento con questo programma bisogna agire *in tempi diversi* per introdurre il testo e per comporlo. È la "composizione asincrona": prima si scrive il testo badando soltanto a contenuto e scansione logica; poi lo si dà "in pasto" a L^AT_EX, che lo compone e lo impagina per produrre il documento finito.

ELENCO DEI SITI INTERNET

CTAN

Sito ufficiale di CTAN.
<http://www.ctan.org/>

EMACS

Pagina Web di Emacs.
<http://www.gnu.org/software/emacs/>

ESCHER

Sito ufficiale di M. C. Escher.
<http://www.mcescher.com/>

FONTS

Catalogo danese dei font per \LaTeX .
<http://www.tug.dk/FontCatalogue/>

GUIT

Sito ufficiale del \GTeX .
<http://www.guitex.org/>

KILE

Pagina Web di Kile.
<http://kile.sourceforge.net/>

KNUTH

Pagina Web di Donald E. Knuth.
<http://www-cs-faculty.stanford.edu/~knuth/>

LAMPOR

Pagina Web di Leslie Lamport.
<http://research.microsoft.com/en-us/um/people/lamport/>

MACTEX

Pagina Web di Mac \TeX .
<http://www.tug.org/mactex/>

MIKTEX

Sito ufficiale di MiK \TeX .
<http://miktex.org/>

SAROVAR

Catalogo \LaTeX online.
<http://texcatalogue.sarovar.org/>

TEXDOC

Versione online di texdoc.
<http://texdoc.net/>

TEXLIVE

Pagina Web di \TeX Live.
<http://www.tug.org/texlive/>

TEXMAKER

Pagina Web di Texmaker.
<http://www.xmlmath.net/texmaker/>

TEXSTUDIO

Pagina Web di TeXstudio.
<http://sourceforge.net/projects/texstudio/>

TEXSHOP

Pagina Web di TeXShop.
<http://www.uoregon.edu/~koch/texshop/>

TEXWORKS

Pagina Web di \TeX works.
<http://www.tug.org/texworks/>

BIBLIOGRAFIA

Beccari, Claudio

- 2012 *Introduzione all'arte della composizione tipografica con L^AT_EX*, <http://www.guitex.org/home/images/doc/guidaguit.pdf>. (Citato alle p. 3, 44, 67.)

Beccari, Claudio e Tommaso Gordini

- 2012 *Codifiche in T_EX e L^AT_EX. Dal sorgente al PDF. Guida pratica per lavorare con successo*, <http://www.guitex.org/home/images/doc/GuideGuIT/introcodifiche.pdf>. (Citato alle p. 13, 21.)

Bringhurst, Robert

- 1992 *The Elements of Typographic Style*, Hartley & Marks, Vancouver, (edizione italiana a cura di Lucio Passerini, *Gli Elementi dello Stile Tipografico*, Sylvestre Bonnard, Milano 2001). (Citato a p. 31.)

Cevolani, Gustavo

- 2006 «Norme tipografiche per l'italiano in L^AT_EX», *ArsT_EXnica*, 1, <http://www.guitex.org/home/images/ArsTeXnica/AT001/arstexnica01.pdf>. (Citato a p. 191.)

De Marco, Agostino e Roberto Giacomelli

- 2011 «Creare grafici con pgfplots», *ArsT_EXnica*, 12. (Citato a p. 149.)

Eco, Umberto

- 1977 *Come si fa una tesi di laurea. Le materie umanistiche*, Bompiani, Milano. (Citato alle p. 131, 132.)

Fairbairns, Robin

- 2012 *The UK T_EX FAQ*, <http://www.tex.ac.uk/tex-archive/help/uk-tex-faq/newfaq.pdf>. (Citato alle p. 32, 66.)

Goossens, Michel, Frank Mittelbach e Johannes Braams

- 2004 *The L^AT_EX Companion*, Addison-Wesley, Reading (Massachusetts). (Citato a p. 19.)

Gregorio, Enrico

- 2009 *Appunti di programmazione in L^AT_EX e T_EX*, 2^a ed., <http://profs.sci.univr.it/~gregorio/introtex.pdf>. (Citato a p. 174.)
- 2010 *L^AT_EX. Breve guida ai pacchetti di uso più comune*, <http://profs.sci.univr.it/~gregorio/breveguida.pdf>. (Citato alle p. 21, 109, 113, 183.)
- 2011 *Introduzione a T_EXworks*, <http://profs.sci.univr.it/~gregorio/introtexworks.pdf>. (Citato a p. 124.)
- 2012 *Installare T_EXLive 2011 su Ubuntu*, <http://profs.sci.univr.it/~gregorio/texlive-ubuntu.pdf>. (Citato a p. 16.)

Guiggiani, Massimo e Lapo Filippo Mori

- 2008 «Consigli su come *non* maltrattare le formule matematiche», *Ars^{TeX}nica*, 5, <http://www.guitex.org/home/images/ArsTeXnica/AT005/arstexnica05.pdf>. (Citato a p. 67.)

Knuth, Donald Ervin

- 1973 *Computer Programming as an Art*, vol. 3, Addison-Wesley, Reading (Massachusetts).
- 1984 *The T_EXbook*, Addison-Wesley, Reading (Massachusetts). (Citato a p. 1.)

Lamport, Leslie

- 1994 *L^AT_EX. A Document Preparation System*, Addison-Wesley, Reading (Massachusetts). (Citato alle p. 3, 19.)

Lesina, Roberto

- 2004 *Il Nuovo Manuale di Stile*, edizione 2.0, con la collaborazione di Federico Boggio Merlo, Zanichelli, Bologna. (Citato alle p. 191, 195.)

Mittelbach, Frank, Gianluca Pignalberi e Dave Walden

- 2007 «Intervista a Frank Mittelbach», *ArsT_EXnica*, 3, <http://www.guitex.org/home/images/ArsTeXnica/AT003/arstexnica03.pdf>. (Citato a p. 6.)

Mori, Lapo Filippo

- 2006 «Tabelle su L^AT_EX 2_ε: pacchetti e metodi da utilizzare», *ArsT_EXnica*, 2, <http://www.guitex.org/home/images/ArsTeXnica/AT002/arstexnica02.pdf>. (Citato a p. 97.)
- 2007 «Scrivere la tesi di laurea con L^AT_EX 2_ε», *ArsT_EXnica*, 3, <http://www.guitex.org/home/images/ArsTeXnica/AT003/arstexnica03.pdf>. (Citato alle p. 35, 113.)

Mori, Lapo Filippo e Maurizio Himmelmann

- 2007 «Scrivere il curriculum vitae», *ArsT_EXnica*, 4, <http://www.guitex.org/home/images/ArsTeXnica/AT004/arstexnica04.pdf>.

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

- 2011 *The Not So Short Introduction to L^AT_EX 2_ε*, <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>. (Citato alle p. 6, 47.)

Pakin, Scott

- 2009 *The Comprehensive L^AT_EX Symbol List*, <http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>. (Citato alle p. 54, 80.)

Pantieri, Lorenzo

- 2007 «Introduzione allo stile ClassicThesis», *ArsT_EXnica*, 6, <http://www.guitex.org/home/images/ArsTeXnica/AT006/arstexnica06.pdf>. (Citato alle p. 30, 31, 57.)
- 2009 «L'arte di gestire la bibliografia con biblalex», *ArsT_EXnica*, 8, <http://www.guitex.org/home/images/ArsTeXnica/AT008/arstexnica08.pdf>.

Pantieri, Lorenzo e Tommaso Gordini

- 2009 *L'arte di fare una presentazione con beamer*, http://www.lorenzopantieri.net/LaTeX_files/Presentazioni.pdf. (Citato a p. 207.)

Sabatini, Francesco e Vittorio Coletti

- 1997 *il Sabatini Coletti. Dizionario della Lingua Italiana*, Giunti Gruppo Editoriale, Firenze.

Serianni, Luca

- 1989 *Grammatica italiana. Italiano comune e lingua letteraria*, con la collaborazione di Alberto Castelvechi, UTET, Torino.

INDICE ANALITICO

COMANDI PARTICOLARI

- `\!`, 75
- `\#`, 23, 24
- `\$`, 23, 24
- `\%`, 23, 24
- `\&`, 23, 24
- `\'`, 54
- `\(`, 65
- `\)`, 65
- `\,`, 69, 74, 75, 190, 198, 200
- `\-`, 50, 190
- `\.`, 54
- `\=`, 54
- `\@`, 51
- `\[`, 66, 92
- `\]`, 25, 47, 49, 56, 60, 62, 98, 106, 111–113, 189
- `\{`, 23, 24
- `\}`, 23, 24
- `\]`, 66, 92
- `\^`, 54
- `_`, 23, 24
- `\'`, 54
- `\|`, 72
- `\~`, 23, 54

- A**
- `\AA`, 54
- `\aa`, 54
- `\abstractname`, 186
- `\addcontentsline`, 38, 40
- `\addplot`, 156–158, 161
- `\addplot/***/`, 158
- `\addplot/*3*/`, 156, 157, 164
- `\addtocategory`, 135
- `\AE`, 54, 204
- `\ae`, 54, 204
- `\aleph`, 80
- `\alert`, 216
- `\alpha`, 71
- `\alsiname`, 186
- Ambiente
 - `abstract`, 34
 - `alertblock`, 218
 - `align`, 38, 81, 82
 - `align*`, 82
 - `aligned`, 82
 - `array`, 91, 92, 94, 99
 - `axis`, 152–154
 - `block`, 216, 218
 - `Bmatrix`, 79
 - `bmatrix`, 79
 - `cases`, 78, 83
 - `CD`, 89
 - `center`, 92, 95
 - `column`, 219, 220
 - `columns`, 219
 - `comment`, 28
 - `description`, 59, 187
 - `document`, 44, 45
 - `doubleSPACE`, 32
 - `enumerate`, 38, 59, 187
 - `eqnarray`, 66
 - `eqnarray*`, 66
 - `equation`, 38, 66, 82
 - `equation*`, 66
 - `exampleblock`, 218
 - `figure`, 38, 93, 116, 152, 183, 219
 - `figure*`, 183
 - `frame`, 212, 219
 - `gather`, 38, 81, 82
 - `gather*`, 82
 - `gathered`, 82
 - `greek`, 209
 - `itaitemize`, 178
 - `itemize`, 59, 187, 212
 - `landscape`, 113
 - `latino`, 203
 - `loglogaxis`, 152
 - `lstlisting`, 139, 140
 - `matrix`, 79
 - `multicols`, 183
 - `multline`, 38, 81
 - `multline*`, 81
 - `onehalfspace`, 32
 - `otherlanguage`, 134
 - `otherlanguage`, 22, 181
 - `otherlanguage*`, 181, 209
 - `pascal`, 148
 - `picture`, 113
 - `pmatrix`, 79
 - `polaraxis`, 152
 - `proof`, 88
 - `quotation`, 61
 - `quote`, 61
 - `quoting`, 61, 62, 181, 191
 - `SCfigure`, 116, 117

- SCfigure*, 117
 - SCTable, 116, 117
 - SCTable*, 117
 - semilogxaxis, 152
 - semilogyaxis, 152
 - sidewaysfigure, 110
 - sidewaystable, 110
 - singlespace, 32
 - smallmatrix, 80
 - smithchart, 152
 - spacing, 32
 - split, 38, 81, 82
 - subequations, 83
 - table, 38, 93, 110, 116, 183, 219
 - table*, 183
 - tabular, 79, 82, 91, 92, 94, 99, 109–111
 - tabularx, 101
 - ternaryaxis, 152
 - thebibliography, 34, 121–123
 - tikzpicture, 152
 - titlepage, 57
 - verbatim, 63, 219
 - verse, 62
 - Vmatrix, 79
 - vmatrix, 79
 - wrapfloat, 119, 120
- A**
- \AmS, 53
 - \and, 27, 56
 - \angle, 80
 - \anwtonos, 207
 - \ap, 54
 - \appendix, 24, 35
 - \appendixname, 186
 - \approx, 81
 - \arccos, 77
 - \arcsin, 77
 - \arctan, 77
 - \arg, 77
 - \arraybackslash, 108
 - \Ars, 23, 53
 - \ast, 80
 - \asymp, 81
 - \author, 27, 214
- B**
- \b, 54
 - \backmatter, 35
 - \bar, 71, 72
 - \beta, 71
 - \bfseries, 55
 - \bibitem, 121
 - \bibname, 123, 186
 - \BibTeX, 53
 - \Big, 77
 - \big, 77
 - \bigcap, 70
 - \bigcup, 70
 - \Bigg, 77
 - \bigg, 77
 - \Biggl, 77
 - \biggl, 77
 - \Biggr, 77
 - \biggr, 77
 - \Bigl, 77
 - \bigl, 77
 - \bigodot, 80
 - \bigoplus, 80
 - \bigotimes, 80
 - \Bigr, 77
 - \bigr, 77
 - \bigskip, 49
 - \binom, 78
 - \blacksquare, 80
 - \bm, 79, 84
 - \bmod, 76
 - \bottomrule, 98
 - \Bra, 78
 - \Braket, 78
 - \bullet, 80
- C**
- \c, 54
 - \cap, 70
 - \caption, 94, 155
 - \captionsetup, 95
 - \ccname, 186
 - \cdot, 80
 - \centering, 95, 107
 - \chapter, 34, 38, 39, 58
 - \chaptername, 186
 - \check, 72
 - \chi, 71
 - \circ, 80
 - \cite, 122, 131, 218
 - \citeauthor, 132
 - \citep, 132
 - \citet, 132
 - \citeyear, 132
- Classe**
- article, 30, 31, 35, 36, 123, 133, 137
 - beamer, 30, 211, 216, 218, 220, 221, 228
 - book, 30, 31, 33, 36, 37, 122, 132, 135, 137
 - KOMA-Script, 30
 - letter, 30
 - memoir, 30

- report, 30, 31, 36, 122, 132, 135, 137
 - suftesi, 30, 181
 - toptesi, 30
 - \cleardoublepage, 123, 193
 - \clearpage, 123, 191, 193
 - \cmidrule, 98
 - \colon, 73
 - \color, 184
 - \columnwidth, 183
 - \complement, 70
 - \cong, 81
 - Contatore
 - secnumdepth, 38, 39
 - tocdepth, 38
 - tocnumdepth, 39
 - \contentsname, 186
 - \cos, 76, 77
 - \cosh, 77
 - \cot, 77
 - \coth, 77
 - \csc, 77
 - \cup, 70
- D**
- \d, 54
 - \dagger, 80
 - \dashv, 80
 - \date, 56, 214
 - \dategreek, 210
 - \ddagger, 80
 - \ddigamma, 207
 - \ddot, 72
 - \ddots, 79
 - \DeclareBibliographyCategory, 135
 - \DeclareMathOperator, 76, 77
 - \defbibheading, 135, 136
 - \deg, 77
 - \Delta, 71
 - \delta, 71
 - \det, 77
 - \dfrac, 85, 106
 - \DH, 54
 - \dh, 54
 - \diamond, 80
 - \Digamma, 207
 - \dim, 77
 - \displaystyle, 85
 - \div, 80
 - \DJ, 54
 - \dj, 54
 - \documentclass, 23, 26, 30
 - \dot, 72
 - \dots, 23, 27, 52, 53, 73, 74, 197
 - \doublespacing, 32
- \Downarrow, 73
 - \downarrow, 73
- E**
- \ell, 80
 - \em, 55, 181
 - \emph, 23, 55
 - \emptyset, 70
 - \enclname, 186
 - \endfirsthead, 112
 - \endfoot, 112
 - \endhead, 112
 - \endlastfoot, 112
 - \epsilon, 71
 - \eqref, 67
 - \equiv, 81
 - \eta, 71
 - \euro, 53, 210
 - \exists, 74
 - \exp, 77
- F**
- \figurename, 186
 - \FloatBarrier, 191
 - \floatname, 185
 - \floatstyle, 185, 186
 - \footcite, 132
 - \footnote, 38, 109
 - \footnotesize, 56
 - \forall, 74
 - \foreignlanguage, 181
 - \frametitle, 212
 - \frenchspacing, 51
 - \frontmatter, 33, 36
 - \fullcite, 132
- G**
- \Gamma, 71
 - \gamma, 71
 - \gb, 178
 - \gcd, 77
 - \ge, 81
 - \gets, 73
 - \gg, 81
 - \glossaryname, 186
 - \graphicspath, 45
 - \greco, 206
 - \Greeknatural, 207
 - \greeknumeral, 207
 - \Grtoday, 210
 - \GuIT, 53
 - \GuIT*, 53

H

$\backslash H$, 54
 $\backslash hat$, 72
 $\backslash hbar$, 80
 $\backslash headtoname$, 186
 $\backslash heartpar$, 182
 $\backslash hline$, 98
 $\backslash hom$, 77
 $\backslash hookleftarrow$, 73
 $\backslash hookrightarrow$, 73
 $\backslash href$, 40
 $\backslash hslash$, 80
 $\backslash hspace$, 106
 $\backslash Huge$, 56
 $\backslash huge$, 56
 $\backslash hypersetup$, 40
 $\backslash hyphenation$, 50

I

$\backslash idotsint$, 69
 $\backslash iff$, 74
 $\backslash iiiint$, 69
 $\backslash iiint$, 69
 $\backslash iint$, 69
 $\backslash Im$, 80
 $\backslash imath$, 80
 $\backslash implies$, 74
 $\backslash in$, 70
 $\backslash include$, 46
 $\backslash includegraphics$, 91, 92, 115, 120, 214
 $\backslash includeonly$, 46
 $\backslash index$, 138
 $\backslash indexname$, 186
 $\backslash inf$, 77
 $\backslash infty$, 69
 $\backslash input$, 45, 46, 207
 $\backslash institute$, 214
 $\backslash int$, 69
 $\backslash iota$, 71
 $\backslash item$, 23, 59, 212, 216, 217
 $\backslash itshape$, 24, 55

J

$\backslash jmath$, 80

K

$\backslash k$, 54
 $\backslash kappa$, 71
 $\backslash katwtonos$, 207
 $\backslash ker$, 77
 $\backslash Ket$, 78

L

$\backslash L$, 54
 $\backslash l$, 54
 $\backslash label$, 39, 67, 88, 94, 119, 121
 $\backslash Lambda$, 71
 $\backslash lambda$, 71
 $\backslash land$, 74
 $\backslash langle$, 90, 197
 $\backslash LARGE$, 56
 $\backslash Large$, 56
 $\backslash large$, 56
 $\backslash LaTeX$, 19, 23, 53
 $\backslash LaTeXE$, 53
 $\backslash latino$, 203, 206
 $\backslash le$, 81
 $\backslash left$, 78
 $\backslash Leftarrow$, 73
 $\backslash leftarrow$, 73
 $\backslash leftleftarrows$, 73
 $\backslash Leftrightarrow$, 73
 $\backslash leftrightarrow$, 73
 $\backslash legend$, 161
 $\backslash lettrine$, 182
 $\backslash lg$, 77
 $\backslash lim$, 77
 $\backslash liminf$, 77
 $\backslash limsup$, 77
 $\backslash linespread$, 24, 32
 $\backslash listfigurename$, 186
 $\backslash listof$, 34, 186
 $\backslash listoffigures$, 34, 119
 $\backslash listoftables$, 34, 119
 $\backslash listtablename$, 186
 $\backslash ll$, 81
 $\backslash ln$, 77
 $\backslash log$, 76, 77
 $\backslash logo$, 214
 $\backslash Longleftarrow$, 73
 $\backslash longleftarrow$, 73
 $\backslash Longleftrightarrow$, 73
 $\backslash longleftrightarrow$, 73
 $\backslash longmapsto$, 73
 $\backslash Longrightarrow$, 73
 $\backslash longrightarrow$, 73
 $\backslash lor$, 74
 $\backslash lstinline$, 139, 140
 $\backslash lstinputlisting$, 139, 140
 $\backslash lstlistoflistings$, 34, 142
 $\backslash lstnewenvironment$, 148
 $\backslash lstset$, 139, 140
 $\backslash lVert$, 72
 $\backslash lvert$, 72

M

`\mail`, 41
`\mainmatter`, 35
`\makeindex`, 137
`\maketitle`, 27, 56
`\MakeUppercase`, 37
`\mapsto`, 73, 74
`\markboth`, 36, 37
`\markright`, 36
`\mathbb`, 70, 84
`\mathbf`, 79, 84
`\mathcal`, 84
`\mathfrak`, 84
`\mathit`, 84
`\mathrm`, 84
`\mathscr`, 84
`\mathsf`, 84
`\mathtt`, 84
`\max`, 77
`\mbox`, 190
`\medskip`, 49
`\MF`, 53
`\mid`, 72, 81
`\midrule`, 98
`\MiKTeX`, 53
`\min`, 77
`\models`, 74
`\MP`, 53
`\mp`, 67
`\mu`, 71
`\multicolumn`, 23, 100

N

`\nabla`, 80
`\ne`, 81
`\nearrow`, 73
`\neg`, 74
`\newblock`, 218
`\newcolumntype`, 107
`\newcommand`, 177, 178
`\newenvironment`, 148, 178, 179
`\newfloat`, 185, 186
`\newline`, 49
`\newpage`, 193
`\newtheorem`, 38, 86, 185
`\newtheorem*`, 86
`\newtheoremstyle`, 87
`\nexists`, 74
`\ni`, 70
`\nobreakspace`, 207
`\nocite`, 131
`\noindent`, 49
`\nonfrenchspacing`, 51
`\normalsize`, 55, 56
`\notag`, 82
`\notin`, 70

`\nu`, 71
`\num`, 200
`\nrightarrow`, 73

O

`\O`, 54
`\o`, 54
`\oddsidemargin`, 32
`\odot`, 80
`\OE`, 54, 204
`\oe`, 54, 204
`\oint`, 69
`\Omega`, 71
`\omega`, 71
`\omissis`, 53, 197
`\onehalfspacing`, 32
`\onslide`, 216
`\oplus`, 80
`\otimes`, 80
`\overbrace`, 78
`\overline`, 71
`\overrightarrow`, 71
`\owns`, 70

P

Pacchetto

`acro`, 64
`acronym`, 64
`afterpage`, 37, 184
`algorithm`, 63
`algpseudocode`, 63
`amscd`, 89
`amsmath`, 38, 53, 65, 76, 77, 81, 82, 99, 106, 210, 211
`amssymb`, 65, 84, 211
`amsthm`, 86, 211
`appendix`, 35
`array`, 101, 106, 107, 109
`ArsClassica`, 181
`babel`, 21, 42, 50, 53, 84, 123, 133, 142, 186, 196, 203–210
`bchart`, 175
`beamer`, 221
`biblatex`, 34, 121, 123–125, 127–129, 131–134, 136, 228
`bm`, 79, 84
`bmpsize`, 114
`bookmarks`, 40
`booktabs`, 91, 98
`braket`, 78
`caption`, 91, 95, 117, 119
`chemfig`, 90
`ClassicThesis`, 30, 57, 181

- crop, 32
- csquotes, 62, 123
- dblfloatfix, 183
- draftwatermark, 184
- dtklogos, 53
- empheq, 85
- emptypage, 37
- epigraph, 181
- etoolbox, 187
- eurosym, 53
- fancyvrb, 63
- float, 34, 38, 96, 183, 185, 186
- fontenc, 20, 42, 182, 209
- fontspec, 209
- footmisc, 58
- fourier, 182
- frontespizio, 57
- geometry, 32
- glossaries, 64
- graphicx, 91, 113, 115, 116, 211
- guit, 53
- hf-tikz, 85
- hologo, 53
- hyperref, 38, 40, 41, 123, 211
- imakeidx, 138
- indentfirst, 49
- inputenc, 13, 20, 42, 209
- LayAureo, 31, 42
- lettrine, 182
- listings, 34, 38, 63, 139, 140, 142, 143, 147, 148
- longtable, 111
- makeidx, 34, 137
- mathpazo, 182
- mathrsfs, 84
- mathtools, 72
- metalogo, 53
- mflogo, 53
- mhchem, 90
- microtype, 49, 209
- minitoc, 37
- mparhack, 57
- multicol, 183
- natbib, 124, 132
- pdflscape, 113
- pgfplots, 174
- PGF/TikZ, 113, 151
- pgfpages, 220
- pgfplots, 151, 152, 154, 156–160, 162, 166, 167, 169, 170, 172, 227
- placeins, 191
- polyglossia, 123, 209
- PSTricks, 113
- pxfonts, 182
- quoting, 61
- ragged2e, 108
- rotating, 110
- setspace, 32
- shapepar, 182
- shorttoc, 37
- showidx, 193
- showkeys, 40
- sidecap, 38, 116, 117
- siunitx, 89, 102, 104, 200, 201
- steroid, 42
- subfig, 38, 117, 118
- supertabular, 111
- tabularx, 100, 101
- teubner, 210
- textcomp, 90
- threeparttable, 109
- TikZ, 151, 152
- tocbibind, 187
- txfonts, 182
- type1ec, 182
- ucs, 20, 124
- url, 41
- varioref, 40
- venndiagram, 89
- verse, 62
- wrapfig, 119
- wrapfloat, 38
- xcolor, 85, 139, 147, 151, 158, 183
- xtab, 111
- xwatermark, 184
- Xy-pic, 113
- xypdf, 89
- \pagebreak, 191, 193
- \pagename, 186
- \pageref, 39
- \par, 47, 55
- \paragraph, 33, 34, 38, 39
- \parallel, 72, 81
- \parencite, 132
- \part, 33, 34, 38, 39
- \partial, 80
- \partname, 186
- \pause, 216
- \pdfbookmark, 23
- \ped, 54, 84
- \permill, 210
- \perp, 81
- \pgfplotsset, 151
- \phantomsection, 38, 123
- \Phi, 71
- \phi, 71
- \Pi, 71
- \pi, 71
- \pianta, 177, 178
- \pm, 67

`\pmod`, 76
`\Pr`, 77
`\prec`, 81
`\preceq`, 81
`\prefacename`, 186, 204
`\printbibliography`, 34, 131, 132, 135, 136
`\printindex`, 34
`\prod`, 68
`\proofname`, 186
`\propto`, 81
`\ProsodicMarksOff`, 203
`\ProsodicMarksOn`, 203
`\Psi`, 71
`\psi`, 71

Q

`\qedhere`, 88
`\qoppa`, 207
`\qqquad`, 44, 68, 75, 106
`\quad`, 44, 74, 75, 106

R

`\r`, 54
`\raggedbottom`, 33
`\RaggedLeft`, 108
`\raggedleft`, 107
`\RaggedRight`, 108
`\raggedright`, 107
`\rangle`, 90, 197
`\Re`, 76, 80
`\ref`, 39
`\refname`, 123, 186
`\renewcommand`, 178, 179
`\renewenvironment`, 179
`\rho`, 71
`\right`, 78
`\Rightarrow`, 73
`\rightarrow`, 73
`\rightrightarrows`, 73
`\rmfamily`, 55
`\rVert`, 72
`\rvert`, 72

S

`\sampi`, 207
`\scriptscriptstyle`, 85
`\scriptsize`, 56
`\scriptstyle`, 85
`\scshape`, 55
`\searrow`, 73
`\sec`, 77
`\section`, 27, 33, 34, 38, 39, 58
`\section*`, 135
`\seename`, 186
`\Set`, 78
`\setbeamercovered`, 214
`\setmainfont`, 209
`\setmainlanguage`, 209
`\setminus`, 70, 72
`\setotherlanguage`, 209
`\setotherlanguages`, 209
`\sffamily`, 55
`\sgn`, 76
`\Sigma`, 71
`\sigma`, 71
`\sim`, 81
`\simeq`, 81
`\sin`, 76, 77
`\spaces`, 32
`\sinh`, 77
`\sisetup`, 102
`\slshape`, 55
`\small`, 24, 56, 139
`\smallskip`, 49
`\sqrt`, 68
`\square`, 80
`\stigma`, 207
`\subfloat`, 23, 117, 119
`\subparagraph`, 34, 38, 39
`\subsection`, 34, 38, 39
`\subsection*`, 135
`\subset`, 70
`\subseteq`, 70
`\subsubsection`, 34, 38, 39
`\succ`, 81
`\succeq`, 81
`\sum`, 68
`\sup`, 77
`\supercite`, 132
`\supset`, 70
`\supseteq`, 70
`\surd`, 80
`\swarrow`, 73

T

`\t`, 54
`\tablename`, 186
`\tableofcontents`, 27, 34, 94
`\tabularnewline`, 108
`\tan`, 77
`\tanh`, 77
`\tau`, 71
`\TeX`, 23, 53
`\text`, 67, 83, 99
`\textbackslash`, 25
`\textbf`, 55, 100, 177
`\textcite`, 132
`\textheight`, 116
`\textit`, 24, 55, 58, 177
`\textrm`, 55
`\textsc`, 55

- `\textsf`, 55
 - `\textsl`, 55
 - `\textstyle`, 85
 - `\textsuperscript`, 54
 - `\texttt`, 55
 - `\textup`, 84
 - `\textwidth`, 32, 92, 101, 116, 183
 - `\tfrac`, 85
 - `\TH`, 54
 - `\th`, 54
 - `\thanks`, 56
 - `\theoremstyle`, 87
 - `\Theta`, 71
 - `\theta`, 71
 - `\tilde`, 72
 - `\tiny`, 56
 - `\title`, 27
 - `\titlegraphic`, 214
 - `\to`, 73, 74
 - `\today`, 23, 24, 204
 - `\toprule`, 98
 - `\triangle`, 80
 - `\triangleleft`, 80
 - `\triangleright`, 80
 - `\ttfamily`, 55, 139
- U**
- `\u`, 54
 - `\underbrace`, 78
 - `\underline`, 71
 - `\underset`, 71
 - `\Uparrow`, 73
 - `\uparrow`, 73
 - `\Updownarrow`, 73
 - `\updownarrow`, 73
 - `\Upsilon`, 71
 - `\upsilon`, 71
 - `\url`, 41
 - `\useoutertheme`, 214
- `\usepackage`, 42
 - `\usetheme`, 214
- V**
- `\v`, 54
 - `\vardigamma`, 207
 - `\varepsilon`, 71
 - `\varphi`, 71
 - `\varpi`, 70, 71
 - `\varrho`, 71
 - `\varsigma`, 70, 71
 - `\vartheta`, 71
 - `\vdash`, 80
 - `\vdots`, 79
 - `\vec`, 71, 72, 79
 - `\vee`, 80
 - `\verb`, 63
 - `\Vert`, 72
 - `\vert`, 71
 - `\vref`, 40
 - `\vspace`, 49
 - `\vspace*`, 49
- W**
- `\wedge`, 80
 - `\widehat`, 72, 73
 - `\widetilde`, 72, 73
 - `\wp`, 80
- X**
- `\XeLaTeX`, 53
 - `\XeTeX`, 53
 - `\Xi`, 71
 - `\xi`, 71
 - `\xleftarrow`, 74
 - `\xrightarrow`, 74
- Z**
- `\zeta`, 71