

LORENZO PANTIERI & TOMMASO GORDINI

L' ARTE

DI SCRIVERE CON

LATEX



Lorenzo Pantieri · Tommaso Gordini

L'arte di scrivere con L^AT_EX

Copyright © 2008-2018

COLOPHON

Questo lavoro è stato realizzato con L^AT_EX su Mac usando ArsClassica, uno stile 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.

Il frontespizio riproduce l'incisione *Tassellazione del piano con uccelli* di Maurits Cornelis Escher (tutte le riproduzioni delle opere di Escher contenute nella guida sono tratte dal sito <http://www.mcescher.com/>).

CONTATTI

✉ lorenzo.pantieri@gmail.com · Scrivi a Lorenzo Pantieri

✉ tommaso.gordini@gmail.com · Scrivi a Tommaso Gordini

La citazione è un utile sostituto dell'arguzia.

– Oscar Wilde

Dedicato a tutti gli appassionati di L^AT_EX.

INDICE

PREFAZIONE	7
INTRODUZIONE	11
1 STORIA E FILOSOFIA	1
1.1 Storia	1
1.2 Filosofia	3
2 INSTALLARE E AGGIORNARE	11
2.1 Ferri del mestiere	11
2.2 Windows	13
2.3 Mac	14
2.4 Linux	14
3 BASI	15
3.1 Per cominciare	15
3.2 Codifiche e lingue	18
3.3 File con cui si ha a che fare	20
3.4 File sorgente	20
3.5 Classi di documento	28
3.6 Gestire la pagina	30
3.7 Strutturare il documento	31
3.8 Stili di pagina	35
3.9 Indice generale, titoli e profondità	36
3.10 Riferimenti incrociati	38
3.11 Collegamenti ipertestuali e indirizzi elettronici	39
3.12 Pacchetti	40
3.13 Unità di misura tipografiche	43
3.14 Documenti di grandi dimensioni	43
4 TESTO	47
4.1 Struttura del testo	47
4.2 Comporre i capoversi	47
4.3 Caratteri particolari e simboli	52
4.4 Modificare stile e corpo del font	54
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
4.9 Acronimi e glossari	64
5 MATEMATICA	65
5.1 Formule in linea e in display	65
5.2 Nozioni introduttive	68
5.3 Operatori	75
5.4 Parentesi	77
5.5 Vettori e matrici	78
5.6 Spezzare formule lunghe	80

5.7	Raggruppare più formule	81
5.8	Modificare stile e corpo del font	83
5.9	Enunciati e dimostrazioni	85
5.10	Altre scienze	88
6	TABELLE E FIGURE	91
6.1	Strumenti fondamentali	91
6.2	Oggetti in testo e fuori testo	92
6.3	Tabelle	97
6.4	Figure	114
6.5	Disposizioni particolari	118
7	BIBLIOGRAFIA	123
7.1	Bibliografia manuale	123
7.2	Bibliografia automatica	125
8	INDICE ANALITICO	139
8.1	Indici semplici	139
8.2	Indici complessi	140
9	PERSONALIZZAZIONI	143
9.1	Comandi e ambienti personali	143
9.2	Personalizzare il testo	145
9.3	Specialità	147
10	REVISIONE FINALE	151
10.1	Problemi orizzontali	151
10.2	Problemi verticali	152
A	NORME TIPOGRAFICHE	157
A.1	Accento e apostrofo	157
A.2	Punteggiatura e spaziatura	158
A.3	Stile del font	160
A.4	Trattamento del testo	161
	ACRONIMI	165
	ELENCO DEI SITI INTERNET	167
	BIBLIOGRAFIA	169
	INDICE ANALITICO	171

PREFAZIONE

La storia di \TeX in Italia è lunga: secondo i resoconti del \TeX Users Group, il primo convegno su \TeX 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 \LaTeX come cominciare a scrivere documenti con questo sistema. Da pochi anni è disponibile la traduzione italiana della *Not So Short Introduction to \LaTeX 2 ε* , che però è ferma a una versione 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 \TeX e \LaTeX .

Ogni utente di \LaTeX 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 *orrori*. Sia chiaro, nessuno è immune dagli *orrori*, compreso chi scrive: quando penso a come scrivevo codice \LaTeX anni fa mi vengono i brividi. Quando leggo una guida a \LaTeX la prima cosa che faccio è cercare gli *orrori* 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 \LaTeX 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 [GIt](#) su questioni stilistiche, su come affrontare problemi tipografici o più strettamente riguardanti \LaTeX . Questa curiosità non è pedanteria fine a sé stessa: è desiderio di esplorare il linguaggio \LaTeX per trovare sempre la soluzione più efficace.

Il risultato è una guida molto piacevole da leggere, che introduce al linguaggio \LaTeX 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 *\LaTeX 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 \LaTeX 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 \LaTeX , 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 L^AT_EX* 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 conquisti sempre nuovi lettori contribuendo alla diffusione di T_EX e L^AT_EX in Italia.

Padova, 30 marzo 2008

Enrico Gregorio

SULLA SECONDA EDIZIONE

La seconda edizione dell'*Arte* ha la grande novità di un secondo autore, Tommaso Gordini, la cui pignoleria linguistica è pari alla pignoleria T_EXnica 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 L^AT_EX con facilità. Gli autori perdoneranno la parola "pignoleria" usata con grande affetto: è caparbietà 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 T_EX è sempre in fermento: X_ET_EX e LuaT_EX 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

SU QUESTA NUOVA EDIZIONE

La guida è stata aggiornata e rivista. L'impianto complessivo dell'opera, come al solito, rimane invariato.

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

Cesena-Padova, 16 aprile 2017

Gli Autori

RINGRAZIAMENTI

Desidero ringrazia-
re innanzitutto i
membri dello staff del [GJIT](#) e poi tutti quelli che hanno
discusso con me sul forum del Gruppo, in particolare Fabia-
no Busdraghi, Gustavo Cevolani, Rosaria D'Addazio, Massimiliano
Dominici, Gloria Faccanoni, Gianluca Gorni, Maurizio Himmelmann, Je-
rónimo Leal, Lapo Filippo Mori, Luigi Scarso, Marco Stara, Andrea Tonelli,
Emiliano Giovanni Vavassori ed Emanuele Vicentini, per l'insostituibile aiu-
to fornito nella redazione di questo lavoro, le spiegazioni dettagliate, la pa-
zienza e la precisione nei suggerimenti, le soluzioni fornite, la competenza
e la disponibilità: grazie mille, ragazzi! Rivolgo un ringraziamento dava-
tro particolare al professor Enrico Gregorio, per i suoi impagabili insegnan-
menti 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 pazien-
za non avrebbe mai raggiunto la forma attuale. Ringrazio infi-
ne 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'ori-
ginale rielaborazione, e Ivan Valbusa, per
i consigli grafici, le idee e per aver
creato lo stile con cui è com-
posta 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, quelle 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 *làtek!*». 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.

Tommaso Gordini

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 note derivano principalmente dalle numerose discussioni presenti sul forum del [GIt](#) (Gruppo Utilizzatori Italiani di \TeX e \LaTeX , [GUI](#)T), 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 essenziali 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 espone alcuni suggerimenti per fare in modo che L^AT_EX produca risultati diversi da quelli predefiniti.

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

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

Questo *non* è un manuale su L^AT_EX, 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 L^AT_EX 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 L^AT_EX 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 contro, 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.

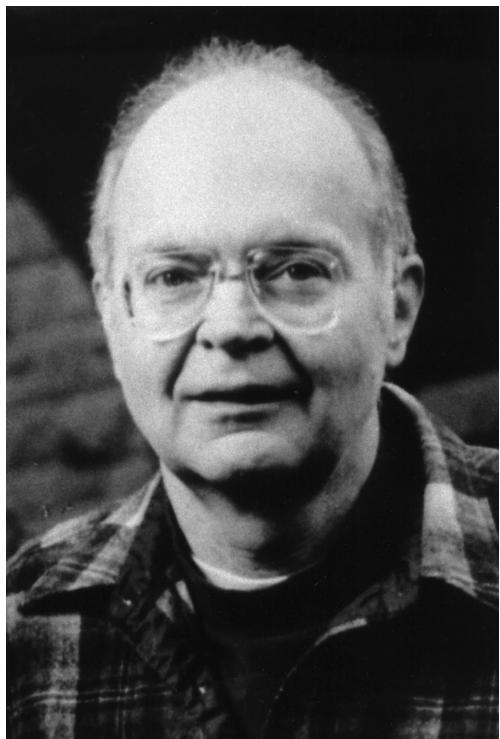


Figura 1: Donald Ervin Knuth

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.

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. 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.

\TeX , rinomato per essere molto stabile e per girare su diversi tipi di calcolatore, è stato rivisto per l'ultima volta nel 2014. Il suo numero di versione converge a π (attualmente è 3,141 592 65).

The \TeXbook [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 *AMS* (American Mathematical Society).

1.1.2 Etimologia

Knuth ha nascosto un trabocchetto nel nome del programma: \TeX , 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: \TeX sarebbe stato il nome perfetto per un programma che compone documenti "allo stato dell'arte".

Knuth dice che «se \TeX 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 *Xopomo* (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 \TeX 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 \LaTeX

\LaTeX (La(mport) \TeX) è un programma di composizione tipografica realizzato da Leslie Lamport e liberamente disponibile, che usa \TeX come motore di tipocomposizione.

\LaTeX non è \TeX , però. Per dare l'idea della differenza tra i due programmi, si potrebbe paragonare \TeX a un corpo, e \LaTeX al più popolare degli "abiti" (fatto, però, di istruzioni in linguaggio \TeX) che nel corso degli anni gli sono stati confezionati addosso per avvicinarlo al pubblico con sembianze amichevoli.

\LaTeX 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*.

Lamport, che collaborava con Knuth allo sviluppo di \TeX , cominciò a scrivere \LaTeX alla fine degli anni Settanta del secolo scorso, quando \TeX non era ancora stato pubblicato.

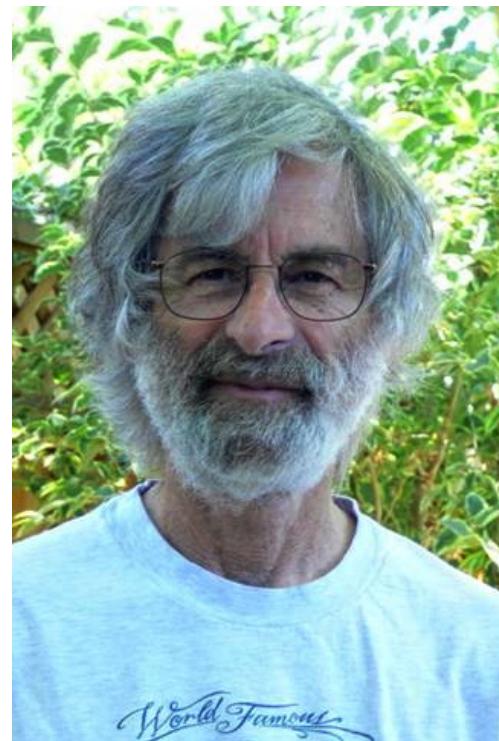


Figura 2: Leslie Lamport

Tabella 1: Cronologia di \TeX e \LaTeX

1977	Knuth comincia a scrivere \TeX
1978	Lamport comincia a scrivere il primo nucleo di \LaTeX
1982	Prima versione pubblica di \TeX
1984	Knuth pubblica la prima edizione di <i>The \TeXbook</i>
1985	Lamport pubblica \LaTeX 2.09 e la prima edizione di <i>\LaTeX. A Document Preparation System</i>
1994	\LaTeX 2 ε
2014	Ultima revisione di \TeX
2016	Ultima revisione di \LaTeX

La prima versione pubblica di \LaTeX 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 programmatore 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 \LaTeX 2 ε , e costituisce l'oggetto di questa guida. *\LaTeX. A Document Preparation System* [Lamport, 1994], scritto da Leslie Lamport, è il manuale ufficiale di \LaTeX .

Il futuro $\text{\LaTeX}3$ si profila come un progetto a lungo termine: i costanti aggiornamenti di \LaTeX , la cui ultima versione pubblica è del 2016, ne costituiscono le tappe di avvicinamento.

1.2 FILOSOFIA

1.2.1 Composizione sincrona e asincrona

La caratteristica che più differenzia \LaTeX dagli altri elaboratori di testo è il fatto che l'introduzione del testo e la sua composizione avvengono *in tempi diversi*.

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 (\LaTeX , in questo caso) che lo impagina solo *successivamente*. L'utente, naturalmente, può modificare il proprio lavoro in qualunque mo-

do 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 Lamport 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 L^AT_EX 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 L^AT_EX, 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 \emph{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.

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

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

$$B = T^{-1}AT \quad (1.1)$$

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 contro

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;
- si occupa dell'impaginazione del documento, mentre l'utente pensa a struttura e contenuto;
- genera strutture complesse come riferimenti incrociati, indici e bibliografie con grande efficienza e flessibilità;
- gestisce in modo impeccabile e tuttora ineguagliato la composizione tipografica di formule matematiche e capoversi;
- è gratuito, multilingue e multiplattaforma;
- ha una struttura modulare che permette di estenderne le capacità per eseguire compiti tipografici non direttamente gestiti dal programma;
- rende davvero difficile creare documenti mal strutturati e tipograficamente scadenti;
- appassiona talmente... che ci si può dimenticare la tazza del caffè sul foglio!

Innegabilmente, L^AT_EX presenta anche alcuni svantaggi.

- Con L^AT_EX ci vuole attitudine all'astrazione.
- Con L^AT_EX la gratificazione non è istantanea, ma ritardata.
- L^AT_EX «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 L^AT_EX 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). L^AT_EX, 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 L^AT_EX decidesse che quella è la soluzione migliore, con le ovvie conseguenze se ciò dovesse accadere durante l'ultimissima revisione del documento.
- L^AT_EX non gestisce pagine più grandi di 33 metri quadrati e non dà denaro alle case di software.

In breve, è opportuno usare L^AT_EX 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 L^AT_EX 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 L^AT_EX 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 Lampert 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{\langle titolo \rangle}
```

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

1.2.6 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.

- *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.
- *Installare una distribuzione di \LaTeX non completa.*
Sebbene possa sembrare inutile occupare diversi gigabyte 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.

- *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 di 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.

- *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.

- *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.

- *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 \Q\!U\!I\!T , 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, Mac e Linux).

2.1 FERRI DEL MESTIERE

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

- un editor di testi con cui scrivere il file sorgente;
- il programma \LaTeX , che lo elabora e produce il documento tipocomposto;
- 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

Anche se oggi molti software di videoscrittura tradizionali prevedono estensioni per trasformare il testo immesso in codice \LaTeX , si consiglia 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.

Gli editor che gestiscono codice \LaTeX sono moltissimi, come si può scoprire con una veloce ricerca in Rete. Quelli consigliati in questa guida sono:

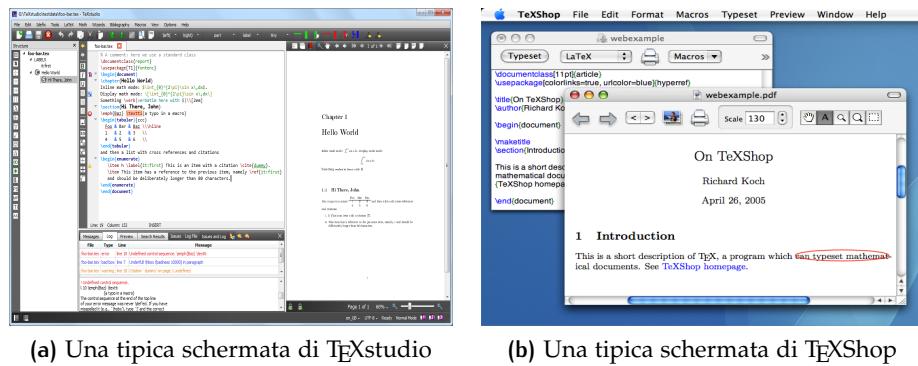


Figura 3: Editor al lavoro

- **TeXstudio** ([TEXSTUDIO](#)), multipiattaforma;
- **TeXShop** ([TEXSHOP](#)), specifico per Mac.

Questi ambienti di lavoro, adatti sia ai principianti che a utenti esperti, fanno parte di **TeX Live** (si veda il paragrafo successivo), e perciò sono già installati sulle macchine (tranne che su Linux).

L’interfaccia grafica di TeXstudio e TeXShop è molto semplice: se ne può vedere un esempio nella figura 3. I programmi sono compatibili con la tecnologia SyncTeX: sorgente e anteprima cioè, sono sincronizzati, e si può “saltare” da un punto dell’uno al corrispondente punto dell’altro (e viceversa).

Per funzionare correttamente, ogni editor deve essere configurato: per esempio, bisogna impostarne la codifica (si veda il paragrafo 3.2.1 a pagina 18) e il motore di composizione, e attivare il controllo ortografico della lingua del documento. Per farlo si consiglia di leggere attentamente la documentazione del proprio editor.

Un pregio di TeXstudio e TeXShop è la capacità di interpretare i *commenti speciali*, cioè “righe magiche” che configurano l’editor in modo automatico, fissando una volta per tutte le impostazioni più importanti del documento. In questo modo si evitano all’utente i problemi descritti nel paragrafo 3.2.1 a pagina 18, e si possono gestire facilmente anche progetti suddivisi in più file come una tesi o un libro. Per maggiori dettagli, si veda [Beccari e Gordini, 2016].

2.1.2 TeX Live

LATeX 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 **LATeX** 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 **LATeX** è CTAN (*Comprehensive TeX Archive Network*, “rete di archivi completi per TeX”, [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. Gli utenti di Windows possono contare sull'alternativa di 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 [18](#)).

Una distribuzione si modifica nel tempo, perché quasi ogni giorno molti dei *pacchetti* che la compongono (si veda il paragrafo [3.12](#) a pagina [40](#)) 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 TeX Live sui diversi sistemi operativi;
- 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

Installare

La procedura descritta di seguito permette di installare TeX Live sul proprio calcolatore.

1. Si scarichi il file `texlive2016.iso` da <http://tug.org/texlive/Images/> e lo si decomprima.
2. Si apra la cartella risultante, si esegua il file `install-tl` al suo interno *come amministratore* e si attenda finché non compare l'omonima finestra.
3. Si seguano le istruzioni sullo schermo senza cambiare nulla e si attenda la fine dell'operazione.

Aggiornare

L'interfaccia grafica di `tlmgr` su Windows è TeX Live Manager.

1. Si avvii il programma (Start → Programmi → TeX Live 2016 → TeX Live Manager).
2. Si prema il pulsante Aggiorna installati e si attenda la fine dell'aggiornamento.

2.3 MAC

Installare

Su Mac si raccomanda di installare \TeX Live tramite Mac \TeX ([MACTEX](#)), un installer che provvede a tutto il necessario.

1. Si scarichi il file Mac \TeX .pkg da <http://www.tug.org/mactex/>.
2. Lo si esegua e se ne seguano le istruzioni fino a completare l'installazione.

Aggiornare

L'interfaccia grafica di `tlmgr` su Mac è \TeX Live Utility.

1. Si avvii il programma (Applicazioni → \TeX → \TeX Live Utility).
2. Si scelga la voce Update All Packages dal menù Actions e si attenda la fine dell'aggiornamento.

2.4 LINUX

A causa di alcuni limiti imposti dagli sviluppatori, non si può installare e aggiornare \TeX Live sulle distribuzioni Linux con la semplicità degli altri sistemi operativi. Le istruzioni descritte nella guida di [Gregorio, 2013] 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 \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):

```
This is pdfTeX, Version 3.14159265-2.6-1.40.17 (TeX Live 2016)
(./primo.tex
LaTeX2e <2016/03/31> patch level 3
Babel <3.9r> and hyphenation patterns for 83 language(s) loaded
(/usr/local/texlive/2016/texmf-dist/tex/latex/base/article.cls
Document Class: article 2014/09/29 v1.4h Standard LaTeX document class
(/usr/local/texlive/2016/texmf-dist/tex/latex/base/fontenc.sty
(/usr/local/texlive/2016/texmf-dist/tex/latex/base/t1enc.def))
(/usr/local/texlive/2016/texmf-dist/tex/latex/base/inputenc.sty
(/usr/local/texlive/2016/texmf-dist/tex/latex/base/utf8.def
(/usr/local/texlive/2016/texmf-dist/tex/latex/base/t1enc.dfu)))
(/usr/local/texlive/2016/texmf-dist/tex/generic/babel/babel.sty
(/usr/local/texlive/2016/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

```
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 gravi 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 (T_EX, 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”);

Tabella 2: 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>)

- 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 quelli consigliati in questa guida), sarà possibile comunicare a L^AT_EX le proprie intenzioni in uno dei modi mostrati nella tabella 2, 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 (T_EX, 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.

ticamente 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 L^AT_EX 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 L^AT_EX i caratteri particolari di una certa lingua già disegnati. Si noti che per funzionare al meglio il pacchetto richiede una distribuzione *completa* di L^AT_EX, altrimenti i caratteri potrebbero apparire "sgranati" sullo schermo (ma non a stampa).
- T1 è la codifica dei font usati 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 L^AT_EX 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 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\u00f2!
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!

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, 2016], 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 L^AT_EX 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 48), date, convenzioni tipografiche e sulla scelta delle regole di sillabazione, che per italiano e principali lingue europee sono già comprese in TeX 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{\langle lingua \rangle}{\langle testo \rangle}
```

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

```
\begin{otherlanguage*}{\langle lingua \rangle}
```

...

```
\end{otherlanguage*}
```

che però non traduce parole fisse e date. Se invece se ne avesse bisogno, si usi l'analogico 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 L^AT_EX 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: L^AT_EX li crea e se ne serve automaticamente.

La tabella 3 a fronte ne raccoglie e descrive i principali.

3.4 FILE SORGENTE

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

3.4.1 Comandi e ambienti

Un comune programma di videoscrittura a composizione sincrona e L^AT_EX presentano per certi versi un funzionamento simile: entrambi ricevono dall'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; L^AT_EX, 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 L^AT_EX di trattare in un certo modo una porzione più o meno ampia di testo. Si possono classificare i comandi di L^AT_EX in base a forma e funzione.

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

Tabella 3: Principali file ausiliari di L^AT_EX

Prodotti da	Estensione	Descrizione
Utente	.bib	Database bibliografico
	.jpg, .pdf, .png	Formati grafici per L ^A T _E X
	.tex	File sorgente
Classi, pacchetti e stili	.bst	Stile bibliografico per Biber
	.cls	Classe di documento
	.sty	Pacchetto
Composizione	.aux	Trasporta informazioni generiche
	.lof	Indice delle figure
	.log	Rendiconta l'ultima composizione
	.lot	Indice delle tabelle
	.toc	Indice generale
Pacchetti e programmi	.bbl	Bibliografia creata con Biber
	.blk	Rendiconto di Biber
	.idx	Voci dell'indice analitico
	.ind	Prodotto di MakeIndex
	.ilg	Rendiconto di MakeIndex
	.out	Segnalibri ipertestuali
Output	.pdf	Prodotto di PDFL ^A T _E X

- 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:

```
\Ars è la rivista del Gruppo
Utilizzatori Italiani di
\TeX e \LaTeX. \\[1ex]
\Ars\ è la rivista del Gruppo
Utilizzatori Italiani di
\TeX{} e \LaTeX.
```

```
\ArsTeXnica è la rivista del Gruppo
Utilizzatori Italiani di \TeX e LATEX.
\ArsTeXnica è la rivista del Gruppo
Utilizzatori Italiani di \TeX 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 4: 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	
		\LaTeX
1		\item{...}
	1	\emph{...}
1	1	\documentclass{...}{...}
2	1	\subfloat{...}{...}{...}
1	2	\pdfbookmark{...}{...}{...}
	3	\multicolumn{...}{...}{...}

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

- Comandi come \textit{\langle testo\rangle} ordinano a L^AT_EX di trattare in un certo modo solo il \langle testo\rangle scritto tra le parentesi graffe.
- Comandi come \itshape, detti *dichiarazioni*, ordinano a L^AT_EX di trattare in un certo modo *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:

Data odierna: \today. \\
Sarò lì in \emph{un} minuto. \\
Tutto il {\itshape testo
seguente è in corsivo}.

Data odierna: 6 gennaio 2018.
Sarò lì in *un* minuto.
Tutto il *testo seguente* è in *corsivo*.

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 4 mostra i principali tipi di comando in base a numero e tipo di argomenti che richiedono.

Tabella 5: Caratteri speciali di L^AT_EX

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

Ambienti

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

```
\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 5. 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:

Tabella 6: 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 + Maiusc + [
}	Alt + 125	⌥ ⇧]	Alt Gr + 8
	Alt Gr + Maiusc +]		Alt Gr + Maiusc +]
~	Alt + 126	⌥ 5	Alt Gr + ì

- i caratteri { ‘ ~ } mancano sulla tastiera italiana: la tabella 6 indica le scorciatoie da prendere in questi casi (su Windows, il codice relativo va digitato *sul tastierino numerico*);
- si distingua 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).

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 43;
- 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 4 nella pagina successiva 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}
```

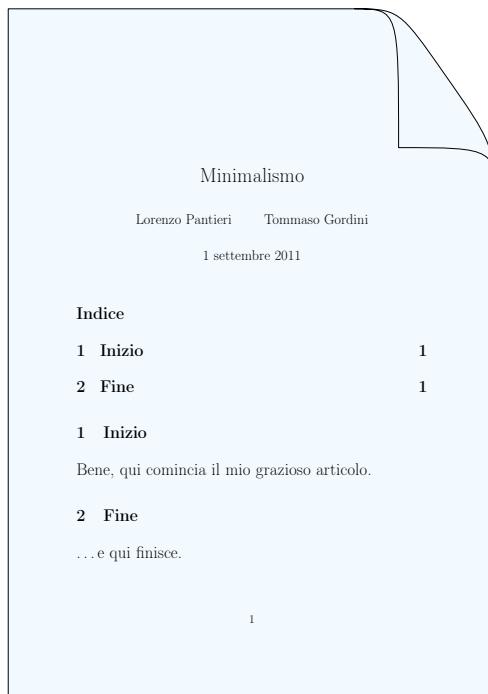


Figura 4: Documento elementare composto con L^AT_EX

```
\begin{document}

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

\tableofcontents

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

\section{Fine}
\ldots 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{\langle titolo \rangle}` 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: Supercalifragili-
stichespiralidoso.

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 L^AT_EX sottovalutano l’importanza di un sorgente “pulito” (con rientri, incolonamenti, 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, incolonamenti, 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
\usepackage[utf8]{inputenc}         % lettere accentate da tastiera
\usepackage[italian]{babel}        % lingua 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 (2017),
\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 di norma come prima cosa 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 sulfesi (per tesi di laurea e dottorato), beamer (per presentazioni). Per esempio, questa guida è scritta con lo stile ArsClassica basato sulle classi KOMA-Script (per maggiori dettagli si veda [Pantieri, 2017]).

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 7 a fronte.

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.

Tabella 7: Opzioni più comuni delle classi standard di L^AT_EX. 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	●	●	●

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 9.3 a pagina 147).

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 10.1 a pagina 151 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.

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 L^AT_EX 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 L^AT_EX 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, 2017]. 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 L^AT_EX 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/retro 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 L^AT_EX come `\textwidth`, `\oddsidemargin`, eccetera, perché la loro azione non tiene in nessun conto le proporzioni di pagina [Fairbairns, 2014].

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 L^AT_EX garantisce un risultato tipografico ottimale e non andrebbe modificata senza una ragione precisa. Per farlo si consiglia il pacchetto `ssetspace`, che modifica lo *scartamento* (o *avanzamento di riga*), ovvero lo spazio fra le righe di base di due righe adiacenti; il pacchetto definisce tre scartamenti *globali* da impostare *nel preambolo* come segue:

- `\singlespacing` (scartamento 1);
- `\onehalfspacing` (scartamento 1,5);
- `\doublespacing` (scartamento 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 lo scartamento per il *<fattore di scala>*.

Riempimento della pagina

In mancanza di istruzioni specifiche L^AT_EX 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 scrivendo nel preambolo `\raggedbottom`.

3.7 STRUTTURARE IL DOCUMENTO

La tabella 8 nella pagina successiva 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.

Tabella 8: 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 che cosa serve	Istruzione	La sezione prodotta ha					Sezione
		numero	titolo	testatine	posto nell'indice		
Corpo del testo	\part \chapter \section \subsection \subsubsection \paragraph \subparagraph	● ● ● ● ● ● ● ○	● ● ● ● ● ● ● ○	● ● ● ● ● ● ● ○	● ● ● ● ● ● ● ○	● ● ● ● ● ● ● ○	Parte Capitolo Paragrafo Sottoparagrafo Sotto-sottoparagrafo Sezione di livello ancora più basso Sezione al più basso livello possibile
Indici	\tableofcontents \listoffigures \listoftables \printindex	○ ○ ○ ○	○ ○ ○ ○	● ● ● ● ● ● ● ○	○ ○ ○ ○ ○ ○ ○ ○	○ ○ ○ ○ ○ ○ ○ ○	Indice generale Elenco delle figure Elenco delle tabelle Indice analitico (makeidx)
Bibliografia	thebibliography \printbibliography	○ ○	● ● ● ● ● ● ● ○	● ● ● ● ● ● ● ○	● ● ● ● ● ● ● ○	● ● ● ● ● ● ● ○	Bibliografia manuale Bibliografia automatica (biblatex)
Varie	abstract	○	●	○	○	○	Sommario

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 titolo 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 10 a pagina 38). Per modificare la numerazione predefinita, si veda il paragrafo 3.9.3 a pagina 37.

3.7.2 Altri sezionamenti

Le altre istruzioni mostrate nella tabella 8 a fronte 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);
- `\mainmatter` (“materiale principale”) numera le sezioni e le pagine con numeri arabi (la numerazione delle pagine riprende da 1);
- `\backmatter` (“materiale finale”) non numera le sezioni e continua la numerazione araba delle pagine dal materiale principale.

Tabella 9: 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> <i>Colophon*</i> <i>Dedica*</i> <i>Sommario*</i> <i>Indice generale*</i> <i>Elenco delle figure*</i> <i>Elenco delle tabelle*</i> <i>Altri elenchi*</i> <i>Prefazione</i> <i>Ringraziamenti*</i> <i>Introduzione non numerata</i>
Materiale principale	<i>Introduzione numerata</i> <i>Capitoli</i> <i>Una o più appendici numerate</i>
Materiale finale	<i>Una o più appendici non numerate</i> <i>Glossario</i> <i>Bibliografia</i> <i>Indice analitico</i>

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 9 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>}
```

*LAT*_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'impone 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 8 a pagina 32. 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{\langle testatina di sinistra\rangle}}%
{\MakeUppercase{\langle testatina di destra\rangle}}
```

che si usa come segue:

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

dove \MakeUppercase produce il proprio argomento in tutte maiuscole (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 8 a pagina 32 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}
```

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 10 nella pagina seguente).

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*

Tabella 10: Corrispondenza fra livelli e sezioni

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

```
\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{\langle etichetta \rangle}
\ref{\langle etichetta \rangle}
\pageref{\langle etichetta \rangle}
```

dove:

- \label assegna agli elementi mostrati nella tabella 11 un'⟨etichetta⟩ arbitraria e *univoca*, che si consiglia di scrivere sempre 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

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

\part	\paragraph*	table	align**	n. di pagina
\chapter	\ subparagraph*	equation	enumerate‡	
\section	\ footnote	multiline**	sidecap†	
\subsection	\ newtheorem	split**	subfig†	
\subsubsection	figure	gather**	wrapfloat†	

* Solo se numerato.

** Richiede il pacchetto amsmath.

‡ Si richiama un elemento della lista.

† Rappresenta l'ambiente o gli ambienti definiti.

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{\langle chiave\rangle=\langle valore\rangle, \dots}

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{\langle indirizzo Internet \rangle}{\langle testo del collegamento \rangle}
```

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://www.guitex.org/}
```

```
http://www.guitex.org/
```

Per i collegamenti a un indirizzo di posta elettronica conviene definire nel preambolo un apposito comando `\mail` (si veda il paragrafo 9.1 a pagina 143),

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

da usare come segue:

```
\mail{lorenzo.pantieri@gmail.com}
```

```
lorenzo.pantieri@gmail.com
```

3.12 PACCHETTI

Scrivendo un documento, prima o poi ci s'imbatte in problemi che L^AT_EX 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 L^AT_EX, 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.

T_EX Live non comprende *tutti* i pacchetti presenti su CTAN. Infatti, compiendo un sorgente può capitare che L^AT_EX 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 12.

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 ne 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 T_EX Catalogue ([TEXCATALOGUE](#)) 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:

- $\langle opzioni \rangle$ è una voce o un elenco di voci separate con la virgola costituite da un solo elemento o un'espressione del tipo $\langle chiave \rangle = \langle valore \rangle$ 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.

- `LATEX` 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 sappendolo 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.

Tabella 12: Principali unità di misura tipografiche riconosciute da L^AT_EX (*m-width* generalmente approssima la larghezza della *M* nel font in uso)

Unità	Codice	Valore
centimetro	cm	
millimetro	mm	
punto	pt	0,3514 mm
<i>x-height</i>	ex	Circa uguale all'altezza della <i>x</i> nel font in uso
<i>m-width</i>	em	Circa uguale al corpo del font in uso

3.12.4 Altri file

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 L^AT_EX su di esso, si ottiene il manuale.
- Eseguendo L^AT_EX 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*. Poiché alcune di esse poco hanno a che fare con quelle più conosciute del sistema metrico decimale, nella tabella 12 si mostrano quelle effettivamente usate in questa guida.

Esistono inoltre i comandi di spaziatura \quad (o *quadrato*) e \quad (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 118).

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` *dove* 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 125), 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` si carcano i file `.tex` in cui si è suddiviso il documento, scrivendo 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* alla cartella dove si trova il file principale (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.

Un metodo alternativo, utile se si devono comporre documenti di grandi dimensioni, è costituito dalla coppia di comandi `\include` e `\includeonly`, la cui sintassi è spiegata in [Beccari, 2016].

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.

Le suddivisioni elencate nella tabella 13 sono fondamentali per comprendere l'articolazione di un testo scritto, e vengono chiamate in generale *sezioni*.

A questo proposito si noti che:

- il sezionamento del documento è compito dell'utente, perché L^AT_EX non lo fa automaticamente;
- è 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 L^AT_EX altrettanto frequentemente si comincia un nuovo capoverso senza nemmeno rendersene conto.

È facile commettere quest'ultimo errore se il testo contiene formule matematiche. 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.

Cominciare un nuovo capoverso

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

- si lascia una riga vuota nel sorgente (di solito si fa così);

Tabella 13: 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

- 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).`

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.)

```
\dots quando Einstein propose
l'equazione
\begin{equation}
E = mc^2
\end{equation}
che è la più nota e la meno
compresa formula della Fisica.
```

```
\dots che, rispetto ai
precedenti, ha alcuni vantaggi.
```

```
La formula
\begin{equation}
D=F-R
\end{equation}
definisce un modello molto
diverso di transistor.
```

```
\dots da cui segue la legge di
Kirchhoff sulle correnti:
\begin{equation}
\sum_{k=1}^n I_k = 0
\end{equation}
```

```
La legge di Kirchhoff sulle
tensioni può essere ricavata\
dots
```

... quando Einstein propose l'equazione

$$E = mc^2 \quad (4.1)$$

che è la più nota e la meno compresa formula della Fisica.

... che, rispetto ai precedenti, ha alcuni vantaggi.

La formula

$$D = F - R \quad (4.2)$$

definisce un modello molto diverso di transistor.

... da cui segue la legge di Kirchhoff sulle correnti:

$$\sum_{k=1}^n I_k = 0 \quad (4.3)$$

La legge di Kirchhoff sulle tensioni può essere ricavata...

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 adeguatamente spaziate tra loro e sillabate a fine riga se proprio non ci stanno, i capoversi presentano la prima riga rientrata per facilitare la lettura. Di solito tutto questo si ottiene dando a mano le rispettive impostazioni; con L^AT_EX, 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{\langle lunghezza \rangle}` 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 L^AT_EX) 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*.

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 `\[\langle lunghezza \rangle]`, in cui *⟨lunghezza⟩* può essere espressa in una qualunque delle unità di misura tipografiche accettate da L^AT_EX, avendo cura di usare *il punto* come separatore decimale (si veda la tabella 12 a pagina 43).

Dividere le parole a fine riga

In generale, L^AT_EX 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 10 a pagina 151 si spiega come fare.

L'algoritmo di sillabazione di L^AT_EX 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 L^AT_EX 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-cro-i-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-istrus-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*. 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 nitro-
idrossilamminico 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.

Il comando

```
\mbox{\langle testo \rangle}
```

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 l'anno avrò imparato il
Fortran. \\[1ex]
Entro l'anno avrò imparato il
\mbox{Fortran}.

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

Spazi interparola e punti fermi

Per giustificare i capoversi L^AT_EX 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 L^AT_EX 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 L^AT_EX 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:

```
Ho studiato il manuale del  
prof. Beccari. \\[1ex]  
Ho studiato il manuale del  
prof.~Beccari.
```

```
Ho studiato il manuale del prof.  
Beccari.
```

```
Ho studiato il manuale del  
prof. Beccari.
```

```
Il tutto è spiegato nel  
paragrafo \ref{sec:par}. \\[1ex]  
Il tutto è spiegato nel  
paragrafo~\ref{sec:par}.
```

```
Il tutto è spiegato nel paragrafo  
4.2.
```

```
Il tutto è spiegato nel paragra-  
fo 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.

Tabella 14: 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	doppie basse	« »	« »
	trattino	-	-
	tratto	--	—
	lineetta	---	—
Puntini	meno	\$-\$	—
		\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 dipendono dalla codifica di input impostata, dall’editor in uso e dalle sequenze di tasti digitate: la tabella 14 raccoglie quelli più usati (scopri l’utente le poche alternative).

Gli esempi seguenti le mostrano all’opera:

Orà è chiaro il concetto di
‘composizione asincrona’.

Ora è chiaro il concetto di ‘compo-
sizione asincrona’.

La Delta di Dirac è una
“funzione impropria”.

La Delta di Dirac è una “funzione
impropria”.

<<Se stai attento, Ermanno,
capisci tutto anche tu.>>

«Se stai attento, Ermanno, capisci
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 14 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 15: Loghi frequenti nel mondo L^AT_EX

Comando	Risultato	Pacchetto richiesto
\TeX	T _E X	
\LaTeX	L ^A T _E X	
\LaTeXe	L ^A T _E X ₂ ε	
\AmS	$\mathcal{A}\mathcal{M}\mathcal{S}$	amsmath
\MF	METAFONT	mflogo
\MP	METAPOST	mflogo
\GuIT, \GuIT*	GUIT, GUIT	guit
\Ars	Ar _S T _E Xnica	guit
\BibTeX	BIBT _E X	dtklogos
\MiKTeX	MiKT _E X	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. L^AT_EX 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 9.1 a pagina 143):

\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 15 elenca i comandi che producono i loghi più comuni nel mondo L^AT_EX (si veda la documentazione dei pacchetti dtklogos e hologo per ulteriori loghi).

L^AT_EX permette di usare accenti e caratteri particolari di molte lingue (nella tabella 16 nella pagina seguente 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!, \'el\`eve,
Sm\o rrebr\o d, z\l otys

Hôtel, naïf, Straße, ¡Señorita!,
élève, Smørrebrød, złotys

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

Tabella 16: Accenti e caratteri particolari

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

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:

- `\ap{\langle testo \rangle}` produce un apice come nelle abbreviazioni oggi in disuso *sig.ra* o *f.lli* (l'alternativa è `\langle testo \rangle`);
- `\ped{\langle testo \rangle}` produce un pedice, utile per qualche sostanza chimica come la vitamina B_{12} , per esempio.

Per l'elenco completo di *tutti* i simboli e i caratteri speciali di \LaTeX (diverse migliaia), si veda [Pakin, 2015].

4.4 MODIFICARE STILE E CORPO DEL FONT

4.4.1 Modificare lo stile

I comandi elencati nella tabella 17 nella pagina successiva 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}}
 maiuscolo corsivo}.`

L'espressione che segue è *in nero corsivo*, ma questa non è *in maiuscolo corsivo*.

A ciascun comando corrisponde una dichiarazione che si comporta come spiegato nel paragrafo 3.4.1 a pagina 20. Anche le dichiarazioni si possono combinare:

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

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

Tabella 17: 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
\emph	\em	Evidenziato
\textit	\itshape	Corsivo
\textsc	\scshape	MAIUSCOLETTTO
\textbf	\bfseries	Nero
\textsl	\slshape	Inclinato
\textrm	\rmfamily	Tondo
\textsf	\sffamily	Senza grazie
\texttt	\ttfamily	Macchina per scrivere

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

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 18 nella pagina successiva 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 è la seconda.

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

Tabella 18: Dichiarazioni per modificare il corpo del font

Dichiarazione	Risultato
\tiny	Esempio
\scriptsize	Esempio
\footnotesize	Esempio
\small	Esempio
\normalsize	Esempio
\large	Esempio
\Large	Esempio
\LARGE	Esempio
\huge	Esempio
\Huge	Esempio

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

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{\langle titolo\rangle}
\author{\langle autore\rangle}
\date{\langle data\rangle}
```

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 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{\langle ringraziamenti\rangle}
```

che nelle classi standard 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 *AMS* \thanks va invece dato in una riga a sé e *fuori* dai comandi.



Figura 5: Esempio d'uso di frontespizio

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 comporre il frontespizio di una tesi di laurea o di dottorato con il pacchetto `frontespizio` (si veda la figura 5). 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 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, 2017].

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 tolto dalla nota e messo nel corpo del testo.

Note a margine

Una nota a margine

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

```
\marginpar{\text{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 `mphack`.)

Note a piè di pagina

Il comando

```
\footnote{\text{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{\text{testo}}
```

che si vede all'opera nell'esempio seguente:

¹ Così.

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

All'interno di un testo già evidenziato, L^AT_EX evidenzia ulteriormente con lo stile tondo.

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:

```
\emph{Fra \textit{un} minuto.} \\  
\textit{Fra \emph{un} minuto.}
```

Fra un minuto.
Fra un minuto.

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.

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}`

Come si può osservare, L^AT_EX evidenzia automaticamente l'argomento del comando secondo le impostazioni generali della classe di documento in uso.

L^AT_EX permette di annidare anche gli elenchi (si consiglia di non annidare 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 L^AT_EX cambia automaticamente il contrassegno negli elenchi anidati, 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

L^AT_EX 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}`

a destra:

Questo testo è allineato a sinistra. L^AT_EX non cerca di creare righe di uguale lunghezza.

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

Questo testo è allineato a
destra. \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 [9.2.2](#) a pagina [147](#) 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 standard `verse`, nel quale:

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

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

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

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* intepretando 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{verb} (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}
```

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

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

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 al pacchetto *listings*.

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).

Il pacchetto *acronym* (se ne veda la documentazione) gestisce 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 L^AT_EX, 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 caricate 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*.

Con L^AT_EX si può scrivere la matematica in due modi: “in linea” e “in display”.

5.1.1 Formule in linea

Una formula “in linea” è incorporata nel testo: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$, per esempio. Come si può osservare, L^AT_EX 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
incorporata nel testo:
`$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$`.
modifica il meno possibile
l’interlinea del capoverso.

Una formula in linea è incorporata nel testo: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$.
L^AT_EX modifica il meno possibile
l’interlinea del capoverso.

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

Ci sono voluti secoli per dimostrare che quando $n > 2$ 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 L^AT_EX 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, 2014].
- 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 L^AT_EX_{2ε} 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:
`\[`
`\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} =`
`\frac{\pi^2}{6}`
`\]`

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

Una formula in display è un’espressione che L^AT_EX 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

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

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

Intorno alla punteggiatura nelle formule in display sono fiorite due scuole di pensiero. Alcuni, tra cui chi scrive, ritengono che non andrebbe *mai* usata, perché superflua e causa di possibili ambiguità nella lettura [Beccari, 2016]. Altri, 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 (tropo 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 38):

```
\begin{equation}
\label{eqn:eulero}
e^{\{i\pi\}}+1=0
\end{equation}
```

Dalla formula `\eqref{eqn:eulero}`
si deduce che \dots

$$e^{i\pi} + 1 = 0 \quad (5.2)$$

Dalla formula (5.2) si deduce
che...

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.

- L^AT_EX 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 24 a pagina 75.
- Nella scrittura delle formule *non sono ammesse righe vuote*.
- L^AT_EX 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:

```
$x+y+z=n$ \\
$ x + y + z = n $
```

$$\begin{aligned} x + y + z &= n \\ x + y + z &= n \end{aligned}$$

```
\[
z^{2+1}=0 \quad \text{per } z=\pm i
\]
```

$$z^2 + 1 = 0 \quad \text{per } z = \pm i$$

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:

```
\[
a^{x+y} \neq a^{x+y}
\]
```

$$a^x + y \neq a^{x+y}$$

5.2.2 Esponenti, indici e radici

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

```
$a_1$ \quad $x^2$ \quad $e^{-\beta t}$ \quad $a_{ij}^3$
```

Il comando `\qquad` produce uno spazio orizzontale di un *quadratone*.

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 `x_n`
estrarre `x_{n_k}`.

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:

```
\[
\sqrt{x} \quad \sqrt[3]{2} \quad \sqrt[\frac{a}{b}]{b}
\]
```

$$\sqrt{x} \quad \sqrt[3]{2} \quad \sqrt[\frac{a}{b}]{b}$$

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.

Trova il massimo della funzione
`\f(x_1, \dots, x_n) = \prod_{k=1}^n x_k`

sotto la condizione
`\sum_{k=1}^n x_k^2 = 1`

Trova il massimo della funzione

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

sotto la condizione

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

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

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

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

```
\[
\frac{1}{x^2+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 ∞ .

```
\[
\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*.

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

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

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.

```
\[
\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 sono disponibili i comandi `\iint`, `\iiint`, `\iiiint` e `\idotsint`.

```
\[
\iint_D f(x,y) dx dy \quad \iiint_D g(x,y,z) dx dy dz
```

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

```
\[
\oint_C f(z) dz = 2\pi i
```

$$\oint f(z) dz = 2\pi i$$

Tabella 19: 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	<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”).

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

I simboli usati nell'esempio precedente sono raccolti insieme ad altri simboli insiemistici nella tabella 19.

Se si scrivono nel preambolo le definizioni seguenti (si veda il paragrafo 9.1 a pagina 143)

```
\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 20 a fronte (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 praticamente 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 9.1 a pagina 143):

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

Tabella 20: Lettere greche

α	<code>\alpha</code>	κ	<code>\kappaappa</code>	ς	<code>\varsigma</code>
β	<code>\beta</code>	λ	<code>\lambdaambda</code>	τ	<code>\tau</code>
γ	<code>\gamma</code>	Λ	<code>\Lambdaambda</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>	ω	<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 *<primo argomento>* rimpicciolito e sovrapposto a quello scritto nel *<secondo argomento>* (di solito un simbolo di relazione binaria), che rimane delle sue dimensioni e nella posizione abituale. Il comando `\underset` fa l'opposto.

Il simbolo

```
\[ \overset{R}{\sim} \]
```

indica un'equivalenza.

Il simbolo

 $\overset{R}{\sim}$

indica un'equivalenza.

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 21 nella pagina seguente).

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

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

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

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

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

\vec{x} \overrightarrow{AB}

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

Tabella 21: 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>

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

$F(x) _{x=\gamma(t)}$	$F(x) _{x=\gamma(t)}$	$ x $
$\$ \lvert x \rvert \$ \\ \lvert x \rvert \$$	$\$ \lvert x \rvert \$$	Se $p \mid n^2$, allora $p \mid n$.

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:

$$\int_a^b f(x) dx = F(x)|_a^b$$

$$\begin{aligned} & \int_a^b f(x) dx = F(x)|_a^b \\ & \int_a^b f(x) dx = F(x)|_a^b \end{aligned}$$

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:

$$A \setminus B$$

$$A \backslash B$$

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:

$$|z| < 1 \quad \|x\| = \sqrt{x_1^2 + \dots + x_n^2}$$

$$\begin{aligned} & |z| < 1 \quad \|x\| = \sqrt{x_1^2 + \dots + x_n^2} \\ & |z| < 1 \quad \|x\| = \sqrt{x_1^2 + \dots + x_n^2} \end{aligned}$$

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

$$\left| \sum_{i=1}^n x_i \right| \quad \left\| \frac{v}{\lambda} \right\|$$

$$\begin{aligned} & \left| \sum_{i=1}^n x_i \right| \quad \left\| \frac{v}{\lambda} \right\| \\ & \left| \sum_{i=1}^n x_i \right| \quad \left\| \frac{v}{\lambda} \right\| \end{aligned}$$

Per aggiungere alle variabili un accento matematico, come un cappello o una tilde, si possono usare i comandi della tabella 21. I comandi `\widehat` e `\widetilde` producono rispettivamente simboli di cappello e tilde che sormontano tutto il proprio argomento (di tre lettere al massimo).

Tabella 22: Frecce

\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>	\Longleftrightarrow	<code>\Longleftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookleftarrow	<code>\hookleftarrow</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>
\leftleftarrows	<code>\leftleftarrows</code>	\rightleftarrows	<code>\rightleftarrows</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>

5.2.9 Punti, frecce e simboli logici

In matematica esistono due tipi di 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:

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

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.

$\begin{bmatrix} x_1, \dots, x_n \end{bmatrix}$	x_1, \dots, x_n
<code>\begin{bmatrix} x_1, \dots, x_n \end{bmatrix}</code>	$x_1 + \dots + x_n$

Per ulteriori esempi dell'uso di \dots, si rimanda al paragrafo 5.5 a pagina 78.

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.

$f: \mathbb{R} \rightarrow \mathbb{R}$	$x \mapsto x^2$
<code>\$f:\mathbb{R}\mathbf{\colon}\mathbb{R}\$ \\</code>	<code>\$x\mapsto x^2\$</code>

I comandi \leftarrow e \rightarrow 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.

Tabella 23: 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>
\Rightarrow	<code>\implies</code>	\iff	<code>\iff</code>	\models	<code>\models</code>

<code>\[</code>	<code>\xleftarrow{a} \quad</code>	<code>\xleftarrow[X]{a+b} \quad</code>	$\xleftarrow[a]{X}$	$\xleftarrow[a+b]{X}$	$\xrightarrow[X+Y+Z]{} \rightarrow$
<code>\]</code>					

La tabella 22 nella pagina precedente raccoglie i principali comandi per ottenere le frecce.

I simboli logici raccolti nella tabella 23 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 L^AT_EX per le formule risulti insoddisfacente. Per modificarla si usano i comandi raccolti nella tabella 24 a fronte. Analogamente a quanto avviene in modo testuale (si veda il paragrafo 3.13), i comandi `\quad` (*quadrato*) e `\quad\quad` (*quadratone*) producono rispettivamente uno spazio di 1 e 2 em.

Spazi troppo stretti

Lo spazio sottile prodotto dal comando `\,` è molto utile in alcune formule. Si confronti

<code>\[</code>	<code>\int_a^b f(x) dx \quad \sqrt{2} a \quad \sqrt{\log x}</code>
<code>\]</code>	

con

<code>\[</code>	<code>\int_a^b f(x) \quad, dx \quad \sqrt{2} a \quad \sqrt{\log x}</code>
<code>\]</code>	

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

<code>\[</code>	$x^{2/2} \quad a/\sin b$	$x^2/2 \quad a/\sin b$
<code>\]</code>		

Tabella 24: Spazi in modo matematico

Comando	Tipo di spazio
\,	Spazio sottile positivo
\!	Spazio sottile negativo
\quad	Spazio di un <i>quadrato</i>
\quad\quad	Spazio di un <i>quadratone</i>

con

<code>\[x^2\!/\!2 \quad a\!\!/\!\!\sin b \]</code>	$x^2/2 \quad a/\sin b$
---	------------------------

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 L^AT_EX, 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;
- i comandi che le producono, come `\sin`, `\cos` e `\log`, sono detti *operatori*.

Li si vede all'opera negli esempi seguenti:

<code>\[\cos 2x \quad \log \log x \quad \log \log(x+y) \]</code>	$\cos 2x \quad \log \log x \quad \log(x + y)$
---	---

Si osservi che:

- nella prima formula, fra cos e 2 c'è più spazio che fra 2 e x ;
- nella seconda i tre elementi sono separati da uno spazio sottile;
- nella terza non c'è alcuno spazio tra log e la parentesi.

Tabella 25: Operatori predefiniti

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

L'unico modo corretto di scriverli è quello appena mostrato: omettendo la barra rovescia si otterebbe “*cos2x*”, tutto in corsivo matematico e senza alcuna spaziatura, che in matematica non significa nulla. Soltanto scrivendo gli operatori come si è appena mostrato L^AT_EX si comporta nel giusto modo e assegna loro font e spazi corretti.

I seguenti sono alcuni operatori predefiniti (la tabella 25 ne riporta l'elenco completo):

$\$\\sin x$, $\\exp x$, $\\log x$,$\\det A$, $\\min_{x \\in A} f(x)$$	$\sin x, \exp x, \log x, \det A,$ $\min_{x \in A} f(x)$
---	--

I due comandi \bmod e \pmod riguardano la relazione di congruenza modulo m :

$\$a\\bmod b\$ \\qquad$ $\$a\\equiv b \\pmod{m}\$$	$a \bmod b \quad a \equiv b \pmod{m}$
---	---------------------------------------

5.3.2 Definire nuovi operatori

Le pubblicazioni specialistiche introducono continuamente nuove funzioni che devono poter essere definite, non essendo contemplate né da L^AT_EX né dai pacchetti dedicati. Risolve il problema il comando \DeclareMathOperator.

Per esempio, per definire la funzione matematica “sgn” che denoti il segno di un numero reale (funzione non prevista né da L^AT_EX né da amsmath), si scrive nel preambolo

```
\DeclareMathOperator{\sgn}{sgn}
```

Nel documento, poi, si darà \sgn per ottenere “sgn” nel font corretto e adeguatamente spaziato su entrambi i lati.

L'operatore che denota la parte reale di un numero complesso è \Re, che produce il simbolo \Re . Lo si può ridefinire perché produca il simbolo in tondo anziché in gotico dando nel preambolo i comandi

```
\DeclareMathOperator{\Realpart}{Re}
\renewcommand{\Re}{\text{\textnormal{Re}}}
```

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.

<pre>\[{a,b,c}\neq{a,b,c} \]</pre>	$a, b, c \neq \{a, b, c\}$
-------------------------------------	----------------------------

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:

<pre>\[\bigl((x-y)+(x+y) \bigr) \]</pre>	$((x - y) + (x + y))$
--	-----------------------

I comandi `\Bigl` e `\Bigr` producono parentesi ancora più grandi:

<pre>\[\Bigl(1+\frac{1}{n}\Bigr)^n \]</pre>	$\left(1 + \frac{1}{n}\right)^n$
--	----------------------------------

I comandi `\biggl` e `\biggr` ne generano di più grandi ancora:

<pre>\[\biggl(\sum_n x_n^2\biggr)^{1/2} \]</pre>	$\left(\sum_n x_n^2\right)^{1/2}$
---	-----------------------------------

Se non basta, ci sono anche `\Biggl` e `\Biggr`. Si noti che alcuni visualizzatori di PDF potrebbero mostrare non correttamente *a schermo* delimitatori particolarmente grandi. Il problema non si presenta, invece, a stampa.

\LaTeX può determinare le dimensioni dei delimitatori anche automaticamente, premettendo `\left` a quello di apertura e `\right` al corrispondente delimitatore di chiusura (se si vuole aprire l'espressione e non chiuderla, si userà `\left.` con il punto finale, e `\right.` in caso contrario). Questi comandi, che funzionano solo se usati in coppia e sulla stessa riga, quasi sempre inseriscono spaziature indesiderate e parentesi più grandi del necessario: perciò va preferito loro il metodo manuale appena descritto.

I comandi `\overbrace` e `\underbrace` creano lunghe graffe orizzontali sopra o sotto un'espressione:

<pre>\[\underbrace{1+2+\dots+n}_{=\frac{n(n+1)}{2}} +(n+1) \]</pre>	$\underbrace{1 + 2 + \cdots + n}_{=\frac{n(n+1)}{2}} + (n + 1)$
--	---

Per scrivere coefficienti binomiali si usa il comando `\binom`:

```
\[
(a+b)^n=
\sum_{\substack{k \in \mathbb{N} \\ 0 \leq k \leq n}}
\binom{n}{k} a^{n-k} b^k
]
```

$$(a+b)^n = \sum_{\substack{k \in \mathbb{N} \\ 0 \leq k \leq n}} \binom{n}{k} a^{n-k} b^k$$

Il comando

```
\substack{\langle sopra \rangle \\ \langle sotto \rangle}
```

produce un indice su più righe.

Per i sistemi di equazioni si può usare l'ambiente `cases` (si veda il paragrafo 5.7.3 a pagina 82):

```
\[
\begin{cases}
x+y=2 \\
x-y=0
\end{cases}
]
```

$$\begin{cases} x + y = 2 \\ x - y = 0 \end{cases}$$

Per gli insiemi è utile il pacchetto `braket`, che definisce l'apposito comando `\Set`:

```
\[
\Set{\frac{1}{n^2} \mid n \in \mathbb{N}}
]
```

$$\left\{ \frac{1}{n^2} \mid n \in \mathbb{N} \right\}$$

e, per le parentesi angolate, `\Bra`, `\Ket` e `\Braket`:

```
\[
\Bra{\psi_n} \quad \Ket{\psi}
\quad \Braket{\psi_n \mid \psi}
]
```

$$\langle \psi_n | \quad | \psi \rangle \quad c_n = \langle \psi_n | \psi \rangle$$

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 `\boldsymbol`; nel terzo il comando `\vec`. Può essere conveniente ridefinire nel preambolo quest'ultimo comando (si veda il paragrafo 9.1 a pagina 143):

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

In questo modo basta scrivere `\vec{v}` per ottenere 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}
```

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

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

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

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}
```

$$A = \begin{bmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Come si può osservare, il comando `\vdots` produce tre punti ellittici verticali e `\ddots` tre in diagonale.

Il comando

```
\hdotsfor{\langle n \rangle}
```

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}
```

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{bmatrix}$$

Una piccola matrice in linea si scrive nell'ambiente `smallmatrix`:

```
Sia $A=\bigl(
\begin{smallmatrix}
a & b \\
c & d
\end{smallmatrix}
\bigr)$
una matrice invertibile.
```

Sia $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ una matrice invertibile.

In questo caso le parentesi vanno aggiunte a mano.

La tabella 26 nella pagina successiva raccoglie altri simboli matematici di uso comune. Per un loro elenco completo si veda [Pakin, 2015].

Tabella 26: Simboli misti (\bigodot, \bigoplus e \bigotimes sono operatori)

\vee	\vee	\wedge	\wedge	\div	\div
\Re	\Re	\Im	\Im	\aleph	\aleph
∂	\partial	∇	\nabla	\cdot	\cdot
\hbar	\hbar	\hslash	\hslash	\circ	\circ
\imath	\imath	\jmath	\jmath	\bullet	\bullet
ℓ	\ell	\wp	\wp	\surd	\surd
\dagger	\dagger	\ddagger	\ddagger	$*$	\ast
\vdash	\vdash	\dashv	\dashv	\angle	\angle
\triangleleft	\triangleleft	\triangleright	\triangleright	\triangle	\triangle
\square	\square	\blacksquare	\blacksquare	\diamond	\diamond
\odot	\odot	\oplus	\oplus	\otimes	\otimes
\bigodot	\bigodot	\bigoplus	\bigoplus	\bigotimes	\bigotimes

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 27 a fronte) nelle formule in linea;
- prima dei simboli di relazione e dopo i simboli binari nelle formule in display;
- *mai* dopo gli operatori funzionali (si veda la tabella 25 a pagina 76), 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

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>	$\begin{aligned} f &= a + b + c \\ &\quad + d + e + g + h \\ &\quad + r + s + t \end{aligned} \quad (5.3)$
--	--

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);

Tabella 27: Simboli di relazione

\leq	<code>\le</code>	\geq	<code>\ge</code>	\sim	<code>\sim</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\cong	<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>
$ $	<code>\mid</code>	\propto	<code>\propto</code>	\equiv	<code>\equiv</code>
\neq	<code>\ne</code>				

- 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 in colonnandole

Per spezzare una formula in più righe incolonnate si usa l'ambiente `split`.

<code>\begin{equation}</code>			
<code>\begin{split}</code>			
<code>a &= b+c-d \\</code>		$a = b + c - d$	
<code>&= e-f \\</code>		$= e - f$	(5.4)
<code>&= g</code>		$= g$	
<code>\end{split}</code>			
<code>\end{equation}</code>			

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

L'ambiente `gather` raggruppa più formule centrandole e numerando automaticamente ciascuna su una riga a sé e, se necessario, assegnandole un'etichetta.

<code>\begin{gather}</code>			
<code>a=b+c \\</code>		$a = b + c$	(5.5)
<code>V+F-S=2</code>		$V + F - S = 2$	(5.6)
<code>\end{gather}</code>			

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 in colonnandole

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 \notag \\
x-1 &= y+z
\end{align}
```

$a = b + c + d$
(5.7)

$e = f$
(5.8)

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 L^AT_EX 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}
```

$a = b$
 $c = d$
 $e = f$
(5.9)

$u = v$
 $w = x$
 $y = z$
(5.10)

Per raggruppare più formule con una parentesi, magari da integrare con del testo esplicativo, L^AT_EX definisce i due ambienti `gathered` e `aligned`. Si veda la documentazione di `amsmath` per la loro sintassi.

5.7.3 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`.

```
\[
\n!=
\begin{cases}
1 & \& \text{se } \$n=0\$ \\
n(n-1)! & \& \text{se } \$n \geq 1 \\
\end{cases}
\]
```

$n! = \begin{cases} 1 & \text{se } n = 0 \\ n(n-1)! & \text{se } n \geq 1 \end{cases}$
(5.11)

L’ambiente `subequations` produce sottonumerazioni:

Tabella 28: Stili dei font matematici (`\mathscr` richiede il pacchetto `mathrsfs`). 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>	$\mathfrak{ABCdef123}$
Nero da lavagna	<code>\mathbb{ABC}</code>	\mathbb{ABC}
Calligrafico	<code>\mathcal{ABC}</code>	\mathcal{ABC}
Manoscritto	<code>\mathscr{ABC}</code>	\mathscr{ABC}

```
\begin{subequations}
\label{eqn:schema}
\begin{aligned}
a &= b+c \\
c &= d \label{eqn:sub} \\
e &= f+g
\end{aligned}
\end{subequations}
Le formule \eqref{eqn:schema},  

e in particolare  

la \eqref{eqn:sub}, \dots
```

5.8 MODIFICARE STILE E CORPO DEL FONT

In modo matematico, L^AT_EX 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 28) agiscano su lettere e numeri, ma *non* sui segni di operazione.

$x+y+2^n M\cos t$ $\mathit{x+y+2^n M\cos t}$ $\mathbf{x+y+2^n M\cos t}$ $\mathsf{x+y+2^n M\cos t}$ $\mathtt{x+y+2^n M\cos t}$ $\mathfrak{x+y+2^n M\cos t}$	$x + y + 2^n M \cos t$ $x + y + 2^n M \cos t$
---	--

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:

```
\[
\mu, M \qquad
\mathbf{\mu}, \mathbf{M}
\]
```

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 `\boldsymbol`.

```
\[
\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}
\]
```

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`.

```
\[
V_{\textup{eff}} \qquad
\psi_{\textup{incidente}}
\]
```

Si confrontino le due scritture seguenti:

V_{eff}	V_{eff}
V_{eff}	V_{eff}

Come si può osservare, L^AT_EX 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 possono impostare a mano la dimensione del font servendosi delle dichiarazioni `\displaystyle`, `\textstyle`, `\scriptstyle` e `\scriptscriptstyle`. Si considerino gli esempi seguenti:

```
\[
\sum_{k=0}^n z^k \qquad
\sum_{k=0}^n z^k \qquad
\sum_{k=0}^n z^k
\]
\$ \displaystyle \sum_{k=0}^n z^k \$ \qquad
\$ \textstyle \sum_{k=0}^n z^k \$ \qquad
\$ \scriptstyle \sum_{k=0}^n z^k \$ \qquad
\$ \scriptscriptstyle \sum_{k=0}^n z^k \$
```

```
\[
x_G =
\frac{\displaystyle \sum_{i=1}^n m_i x_i}{\displaystyle \sum_{i=1}^n m_i}
\]
```

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{...}`:

<pre>\$\frac{1}{n} \log x \qquad \frac{1}{n} \log x \\$ \dfrac{1}{n} \log x \qquad \dfrac{1}{n} \log x \[\frac{1}{n} \log x \qquad \frac{1}{n} \log x \] \frac{1}{n} \log x \qquad \frac{1}{n} \log x</pre>	$\frac{1}{n} \log x \qquad \frac{1}{n} \log x$ $\dfrac{1}{n} \log x \qquad \dfrac{1}{n} \log x$ $\left[\frac{1}{n} \log x \qquad \frac{1}{n} \log x \right]$
--	---

5.9 ENUNCIATI E DIMOSTRAZIONI

Di qui in avanti si dà per caricato *anche* il pacchetto `amsthm`.

5.9.1 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 (`definizione` e `teorema`, per esempio; non si può usare `def`, perché è già un comando di base di L^AT_EX);
- 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` mette a disposizione 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 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}
```

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.
```

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.

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, L^AT_EX:

- 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:

```
\begin{legge}
\label{lex:capo}
Il capo ha ragione.
\end{legge}
\begin{decreto}[Aggiornamento
alla legge~\ref{lex:capo}]
Il capo ha \emph{sempre} ragione.
\end{decreto}
\begin{legge}
Se il capo ha torto, vedere la
legge~\ref{lex:capo}.
\end{legge}
```

Legge 1. *Il capo ha ragione.*

Decreto 2 (Aggiornamento alla legge 1). *Il capo ha sempre ragione.*

Legge 3. *Se il capo ha torto, vedere la legge 1.*

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:

```
\begin{murphy}
Se qualcosa può andar male,
lo farà.
\end{murphy}
```

Murphy 5.9.1. *Se qualcosa può andar male, lo farà.*

5.9.2 Dimostrazioni

L'ambiente *proof* permette di scrivere una dimostrazione, che nel documento finito sarà chiusa da un quadratino. Si osservi l'esempio seguente:

```
\begin{teorema}[di Euclide]
I numeri primi sono infiniti.
\end{teorema}
\begin{proof}
Per assurdo, sia dato l'elenco
finito di tutti i primi. Siano
$q$ il loro prodotto e $p$ un
primo che divide $q+1$. Allora
$p$ non sta nell'elenco, perché
altrimenti dividerebbe $q+1$.
\end{proof}
```

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}
```

con

```
\begin{proof}
Basta usare la formula
\[
E=mc^2 \qedhere
\]
\end{proof}
```

Dimostrazione. Basta usare la formula

$$E = mc^2$$

□

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.

5.10 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 finito operando un'unica modifica nel preambolo anziché agire a mano su ciascuna unità di misura.

Dato che le convenzioni tipografiche italiane prevedono la virgola e non il punto (predefinito dal pacchetto) come separatore decimale, il pacchetto va caricato almeno con l'opzione seguente:

```
\usepackage[output-decimal-marker={,}]{siunitx}
```

Ecco qualche esempio:

```
\SI{23.4}{kg.m.s^{-2}} \\
\$r=\SI{0.8768(11)e-15}{m}\$ \\
\si{\joule\per\mole\per\kelvin} \\
\si{j.mol^{-1}.K^{-1}}
```

$23,4 \text{ kg m s}^{-2}$
 $r = 0,8768(11) \times 10^{-15} \text{ m}$
 $\text{J mol}^{-1} \text{ K}^{-1}$
 $\text{j mol}^{-1} \text{ K}^{-1}$

<code>\SI{100}{\celsius} \\ \ang{1;2;3}</code>	100 °C 1°2'3"
--	------------------

Tutti i precedenti comandi funzionano sia in modo testuale sia in modo matematico. Si tenga presente, tuttavia, che se le unità di misura da inserire sono poche, si possono scrivere più semplicemente così:

L'auto viaggia a $\sim 65 \text{ km/h}$.	L'auto viaggia a 65 km/h.
---	---------------------------

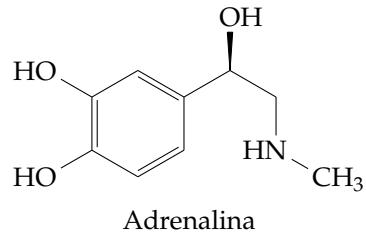
Se si prevede di inserire i simboli dei gradi Celsius o di quelli sessualiimali, si consiglia di caricare *anche* il pacchetto `textcomp`, che abilita la ricerca di $^\circ$ nel documento finito.

Per comporre semplici formule chimiche è utile il pacchetto `chemformula`.

<code>\ch{H2O} \qquad</code> <code>\ch[^{227}_{-90}Th+} \qquad</code> <code>\ch{C6H5-CHO} \\ [2ex]</code> <code>\ch{SO4^2- + Ba^2+ -> BaSO4 v}</code>	H_2O $^{227}_{90}\text{Th}^+$ $\text{C}_6\text{H}_5\text{-CHO}$ $\text{SO}_4^{2-} + \text{Ba}^{2+} \longrightarrow \text{BaSO}_4 \downarrow$
---	---

Il pacchetto `chemfig` compone strutture molecolari, formule di reazione e tutta la chimica di cui si può aver bisogno.

<code>\chemname{\chemfig{*6((-HO)-=%</code> <code>-(-(<[:60]OH)-[:-60]-%</code> <code>[:::-60,,,2] HN-[:::+60]CH_3)=%</code> <code>-(-HO=)})}{Adrenalina}</code>	
--	--



Chi si occupa di informatica ha bisogno delle parentesi angolate $\langle \dots \rangle$, che si ottengono con i comandi `\langleangle` e `\rangleangle` in modo matematico.

Per la scrittura di tutte le altre discipline scientifiche si rimanda al catalogo tematico dei pacchetti disponibile su [TEXCATALOGUE](#).

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 L^AT_EX 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 di tre alcaloidi.

Alcaloide	Origine
atropina	belladonna
morfina	papavero
nicotina	tabacco

mostra l’origine di tre alcaloidi.

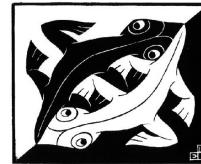
La tabella
\[
\begin{array}{ll}
\toprule
f(x) & f'(x) \\
\midrule
x^n & nx^{n-1} \\
e^x & e^x \\
\sin x & \cos x \\
\bottomrule
\end{array}
\]
mostra le derivate di alcune funzioni elementari.

$f(x)$	$f'(x)$
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 9.3.1 a pagina 147). 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 lì*, perché richiesto dal discorso, una figura alta cinque centimetri avendone però soltanto tre a disposizione: va da sé che *lì* la figura *non* ci può stare in nessun modo: se lo spazio fisico non c'è, non lo si può inventare! Questa situazione, si badi bene, non si verifica solo con L^AT_EX, 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 lì” 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 L^AT_EX 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 L^AT_EX.

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 10.2 a pagina 152 si mostra che cosa (eventualmente) si può fare durante la revisione per risolvere collocazioni poco gradite e migliorarle ulteriormente.

Ambienti standard per gli oggetti mobili

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 38).

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 *dove* essere messa anche nel sorgente.

Le classi standard prevedono la didascalia sempre *sotto* l’oggetto, perciò L^AT_EX 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.

Tabella 29: Preferenze di collocazione per gli oggetti mobili

Preferenza	Chiede a L ^A T _E X 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

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 (L^AT_EX centra automaticamente le didascalie che occupano una sola riga);
- `labelfont={sf, bf}` imposta l'etichetta della didascalia in caratteri senza grazie neri.

Che cosa fare durante la stesura

Durante la stesura del documento, si sa, bisogna concentrarsi sul contenuto del lavoro, lasciando fare a L^AT_EX tutto il resto. La gestione degli oggetti non si sottrae a questa regola, 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 L^AT_EX), 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 L^AT_EX, questo, ma un vantaggio, proprio come lo è il non doversi preoccupare di numerare a mano sezioni e pagine o il non dover pensare a quanto spazio ci va tra un titolo e il testo successivo.

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 29, che “suggeriscono” a L^AT_EX come si vorrebbero vedere gli oggetti sulle pagine del proprio documento. L^AT_EX seguirà i suggerimenti nell'ordine in cui li trova (fa eccezione la preferenza p, come si spiega nel paragrafo 10.2 a pagina 152). 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 L^AT_EX 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).

`\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 6: Figura collocata a mano

La figura 6 è un esempio di figura mobile collocata a mano.

6.3 TABELLE

Comporre tabelle di altissima qualità è una delle specialità di L^AT_EX, 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;

Tabella 30: Tabella che non rispetta le regole generali

<i>D</i>	<i>P</i>	<i>u</i>	β	<i>G</i>
.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 31: Tabella che rispetta le regole generali

<i>D</i> (m)	<i>P</i> (kg)	<i>u</i> (m)	β	<i>G</i> (Pa)
0,500	269,8	0,000674	1,79	0,04089
1,50	421,0	0,001035	3,59	0,04089
10,0	640,2	0,001565	7,18	0,04089

- 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 162).

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 [30](#) e [31](#).

Separare le celle e chiudere le righe

Le celle di una tabella vanno separate tra loro con il carattere separatore & 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

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);

Tabella 32: 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

3. `\bottomrule` produce l'ultimo filetto.

Il comando

`\cmidrule{⟨n⟩-⟨m⟩}`

invece, disegna un filetto orizzontale dalla sinistra della colonna *⟨n⟩*-esima fino alla destra della colonna *⟨m⟩*-esima. Per migliorare leggermente la resa della tabella, si può specificare subito dopo `\cmidrule` un argomento facoltativo *⟨troncamento⟩* *fra parentesi tonde*:

`\cmidrule(⟨troncamento⟩){⟨n⟩-⟨m⟩}`

che accetta tre possibilità: r, l o rl. Si sta dicendo a L^AT_EX 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): potrebbe servire se lo spazio è poco o semplicemente come tocco di stile.

Codice ordinato

Anche se L^AT_EX 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.

6.3.2 Tabelle standard

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 & & newton \\
energia & & joule \\
tensione & & volt \\
\bottomrule
\end{tabular}
```

Grandezza	Simbolo	Unità
forza	F	newton
energia	E	joule
tensione	V	volt

```
\[
\begin{array}{cc}
\texttt{\toprule} \\
f(x) & \text{& } \texttt{\text{Una primitiva}} \\
\texttt{\midrule} \\
e^x & e^x \\
\cos x & \sin x \\
\sin x & -\cos x \\
\texttt{\bottomrule} \\
\end{array}
\]
```

$f(x)$	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 32 nella pagina precedente;
- in `tabular`, eventuali formule matematiche si scrivono con i comandi per le formule in linea, per esempio fra dollari $\$ \dots \$$;
- 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

`*{\langle n \rangle}{\langle descrittore \rangle}`

equivalente a $\langle n \rangle$ colonne consecutive del tipo indicato dal *descrittore*.

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 L^AT_EX 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{\langle larghezza \rangle}`, 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 105.

Colonne di larghezza prefissata

Il codice seguente, che mostra all’opera il descrittore `p{\langle larghezza \rangle}`, produce la tabella 33 a fronte:

Tabella 33: 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.

```
\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 (per forzare l'andata a capo in una cella si usa `\newline`);
- 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.

Tabelle di larghezza prefissata

Il pacchetto tabularx, che carica il pacchetto array (si veda il paragrafo 6.3.8 a pagina 108), definisce l'omonimo ambiente tabularx e un nuovo tipo di colonna X, che alle caratteristiche delle colonne p appena esaminate aggiunge un vantaggio: è L^AT_EX 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 34 nella pagina successiva, identica alla 33 nel contenuto, ottenuta con il codice seguente:

```
\begin{tabularx}{\textwidth}{lX}
\toprule
\textbf{Forza} & Una forza è una grandezza fisica che si manifesta
```

Tabella 34: 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 35: Tabella con due colonne 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 altopiani estesi.	Fauna: si diffondono i rettili; nei mari prosperano pesci e invertebrati. Flora: si sviluppano alghe caratteristiche.

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 35, prodotta con il codice seguente, mostra come a *tutte* le colonne X, se più d'una, L^AT_EX assegna 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, 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}
```

Tabella 36: Tabella con colonna S

Espressione	Valore
π	3,1416
π^π	36,46
π^{π^2}	80 662,7

6.3.4 Colonne di soli numeri

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 *specifica 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 162);
- 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 162).

Lo si vede all'opera nella tabella 36, 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^2}}\$ & & 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.

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:

Tabella 37: Tabella con cella multicolonna

Nome Particella	Nome Antiparticella	Carica (e)
elettrone	positrone	∓ 1
protone	antiprotono	± 1
neutrone	antineutrone	0

<code>\begin{tabular}{S}</code>	<code>\toprule</code>	<code>\midrule</code>	<code>\bottomrule</code>	<code>\end{tabular}</code>
<code>\\$P\\$ & \\$u\\$ \\</code>				
<code>\{(kg\} & \{(m\} \\</code>				
<code>\midrule</code>				
<code>269.8 & 0.000674 \\</code>				<code>P</code>
<code>421.0 & 0.001035 \\</code>				<code>(kg)</code>
<code>640.2 & 0.001565 \\</code>				<code>u</code>
				<code>(m)</code>
				<code>269,8</code>
				<code>0,000 674</code>
				<code>421,0</code>
				<code>0,001 035</code>
				<code>640,2</code>
				<code>0,001 565</code>

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:

<code>\begin{tabular}{%</code>	<code>S[table-format=3.1]%</code>	<code>S[table-format=1.6]}</code>	<code>\toprule</code>	<code>\midrule</code>	<code>\bottomrule</code>	<code>\end{tabular}</code>
<code>\\$P\\$ & \\$u\\$ \\</code>						
<code>\{(kg\} & \{(m\} \\</code>						
<code>\midrule</code>						
<code>269.8 & 0.000674 \\</code>					<code>P</code>	
<code>421.0 & 0.001035 \\</code>					<code>(kg)</code>	
<code>640.2 & 0.001565 \\</code>					<code>u</code>	
					<code>0,000 674</code>	
					<code>0,001 035</code>	
					<code>0,001 565</code>	

Nello stesso modo è stata prodotta la tabella 31 a pagina 98.

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 38: Tabella con celle multiriga

Famiglia	Particella	Simbolo
leptoni	elettrone	e
	muone	μ
	tau	τ
	neutrino	ν
barioni	protone	p
	neutron	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, si unita 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{\langle n \rangle}{\langle descrittore \rangle}{\langle testo \rangle}
```

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 37 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{\langle n \rangle}{\langle testo \rangle}
```

Tabella 39: Tabella con celle multiriga e multicolonna

Elemento	Strati		
	K	L	M
idrogeno	1		
litio	2	1	
sodio	2	8	1

(che richiede il pacchetto omonimo) crea una cella alta $\langle n \rangle$ righe, il cui $\langle testo \rangle$ verrà centrato verticalmente. Il codice

```
\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 38 nella pagina precedente. 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 39.

6.3.7 Spaziare a mano righe e colonne

Quando (molto di rado) i risultati di L^AT_EX 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

```
\[\langle altezza\rangle]
```

che, sostituito al delimitatore di riga standard `\`` nel corpo della tabella, abbas-
sa la riga immediatamente successiva (e solo quella) di uno spazio verticale
pari ad `\langle altezza\rangle`. Di solito s'inserisce questo spazio supplementare per ri-
mediare a lievi sovrapposizioni (specialmente di formule matematiche in
display) oppure per dare maggiore “respiro” alla tabella come si è fatto nel-
la tabella 43 a pagina 112, dove si sono allontanati leggermente i blocchi di
testo altrimenti troppo vicini.

Si osservi come la spaziatura automatica nella tabella seguente è troppo
risicata

```
\[
\begin{array}{cc}
\toprule
f(x) & f'(x) \\
\midrule
\log x & \frac{1}{x} \\
\arctan x & \frac{1}{1+x^2} \\
\bottomrule
\end{array}
\]
```

$f(x)$	$f'(x)$
$\log x$	$\frac{1}{x}$
$\arctan x$	$\frac{1}{1+x^2}$

e come la si può correggere molto semplicemente con

```
\[
\begin{array}{cc}
\toprule
f(x) & f'(x) \\
\midrule
\log x & \frac{1}{x} \\
\arctan x & \frac{1}{1+x^2} \\
\bottomrule
\end{array}
\]
```

$f(x)$	$f'(x)$
$\log x$	$\frac{1}{x}$
$\arctan x$	$\frac{1}{1+x^2}$

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 larghezza\rangle`. Si
noti che in questo caso la `\langle larghezza\rangle` va indicata:

- mettendo le consuete espressioni (valore e unità di misura) nell'argo-
mento di `\hspace{\langle ... \rangle}`;
- con comandi di spaziatura fissa (`\quad` e `\quadquad`, 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 $\{>\langle dichiarazione\ iniziale\rangle\}$ e $\{<\langle dichiarazione\ finale\rangle\}$, la cui sintassi completa è

```
>\{<\langle dichiarazione\ iniziale\rangle\}\langle descrittore\rangle\{<\langle dichiarazione\ finale\rangle\}
```

dove:

- $\{>\langle dichiarazione\ iniziale\rangle\}$ va data prima del $\langle descrittore\rangle$ per dire a L^AT_EX di eseguirla *prima* del contenuto della colonna;
- $\langle descrittore\rangle$ è un descrittore di colonna, di quelli già esaminati o eventualmente definito con \newcolumntype (si veda più sotto);
- $\{<\langle dichiarazione\ finale\rangle\}$ va data dopo il $\langle descrittore\rangle$ 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 \newcolumntype, la cui sintassi completa è

```
\newcolumntype{\langle carattere\rangle}{>\{<\langle dichiarazione\rangle\}\langle descrittore\rangle\{<\langle dichiarazione\rangle\}}
```

dove $\langle carattere\rangle$ 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 $\langle dichiarazione\ iniziale\rangle$ una delle dichiarazioni elencate nella tabella 17 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 33 a pagina 101, prodotta in quella sede specificando a mano lo stile della prima colonna.

Tabella 40: Tabelle con testo diversamente allineato in una colonna p

(a) Testo allineato a destra.	(b) Testo allineato a sinistra.		
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.

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 *<dichiarazione iniziale>* 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*.

Si confrontino le tabelle 40a e 40b, 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

Tabella 41: Tabella ottenuta con array

$\int \cos x \, dx$	$\sin x + c$
$\int e^x \, dx$	$e^x + c$
$\int \sec^2 x \, dx$	$\tan x + c$

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 sillabarla, 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

```
\newcolumntype{L}{>{$}l<{$}}
\newcolumntype{C}{>{$}c<{$}}
\newcolumntype{R}{>{$}r<{$}}
```

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

```
\newcolumntype{L}{>{$\displaystyle l<{$}}
\newcolumntype{C}{>{$c<{$}}
\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 41.

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].

Tabella 42: 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.

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 L^AT_EX, 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.

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 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 18 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 42. (Se ne confronti il risultato con la tabella 33 a pagina 101.)

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 8 a pagina 32. Per ruotare le immagini il pacchetto definisce l'analogo ambiente `sidewaysfigure`.

Tabelle su più pagine

Una tabella *deve* sempre poter stare in una sola pagina: se è più lunga, L^AT_EX 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`), `longtable` di solito 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 43, 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 43: 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>).

Continua nella prossima pagina

Continua dalla pagina precedente

Comando	Effetto
<code>[⟨carattere⟩]</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>{⟨preambolo⟩}</code>	È il consueto preambolo nel quale indicare i descrittori già esaminati nei paragrafi precedenti.
<code>\caption{⟨didascalia⟩} \\</code>	Se specificato, assegna alla tabella una didascalia, anche nella versione <code>[⟨didascalia breve⟩]</code> (si veda il paragrafo 6.2.2 a pagina 93).
<code>\label{⟨etichetta⟩} \\</code>	Assegna alla tabella un'etichetta, utile per i riferimenti incrociati (si veda lo stesso paragrafo).
<code>⟨intestazione iniziale⟩ \\</code> <code>\endfirsthead</code>	Specifica l' <code>⟨intestazione iniziale⟩</code> , cioè l'intestazione della tabella nella prima pagina in cui compare.
<code>⟨intestazione normale⟩ \\</code> <code>\endhead</code>	Specifica l' <code>⟨intestazione normale⟩</code> , 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 <code>⟨piede normale⟩</code> , 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 <code>⟨piede finale⟩</code> , 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 43 a fronte:

```
\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 L^AT_EX, tanto che ne esistono di specifiche, cui si rimanda per gli approfondimenti.

Questi oggetti presentano almeno due problemi diversi, riguardanti:

- il *tipo di file* da introdurre nel documento (verrà trattato in questo paragrafo);
- 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 schemi e grafici, argomento non considerato in questa guida per via della sua complessità (gli strumenti più diffusi per disegnare con \LaTeX sono i pacchetti *TikZ*, che produce disegni d'ogni tipo, e *pgfplots*, che fa grafici di funzioni). 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 successiva 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 rimpicciolate 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 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 *bmpsize*, infine, permette di inserire anche immagini *BMP*, *GIF* e *TIFF*.

Il paragrafo 6.4.4 nella pagina seguente 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

Tabella 44: 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
Adobe Acrobat			•	•
Adobe Reader				•
Anteprima		•	•	•
Asymptote	•			
Ghostview e GSview		•		•
GIMP		•	•	•
Gnuplot	•			
ImageMagick			•	
Inkscape		•	•	
Mathematica			•	
OmniGraffle		•		
Xfig e WinFIG	•			

(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 44 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>}
```

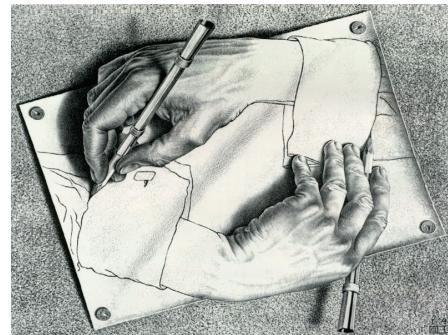
le include nel documento. Si osservi che:

- nell'argomento facoltativo ci vanno le opzioni che regolano l'aspetto della figura sulla pagina nella forma `<chiave>=<valore>` (si veda la tabella 45 a pagina 118);
- 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}
```



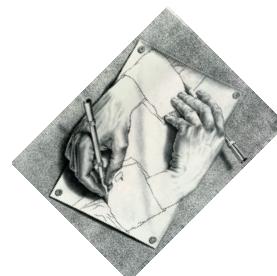
Ancora, la può ridimensionare nel suo complesso

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



o ruotare di un certo angolo in entrambi i sensi:

```
\includegraphics[width=0.5%
\textwidth,angle=45]{mani}
```



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.

Tabella 45: Principali chiavi di graphicx

Chiave	Agisce su
width	Larghezza
height	Altezza
scale	Larghezza e altezza
angle	Orientamento

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

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}[\langle larghezza relativa \rangle][\langle collocazione \rangle]`

e

`\begin{SCtable}[\langle larghezza relativa \rangle][\langle collocazione \rangle]`

dove:

- $\langle \text{larghezza relativa} \rangle$ indica il rapporto fra larghezza della didascalia e larghezza della figura (o della tabella) corrispondente. Un valore elevato di $\langle \text{larghezza relativa} \rangle$ (50, per esempio) assegna alla didascalia la massima larghezza possibile.
- $\langle \text{collocazione} \rangle$ 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 7 a fronte è stata inclusa nel documento con un codice del tipo:

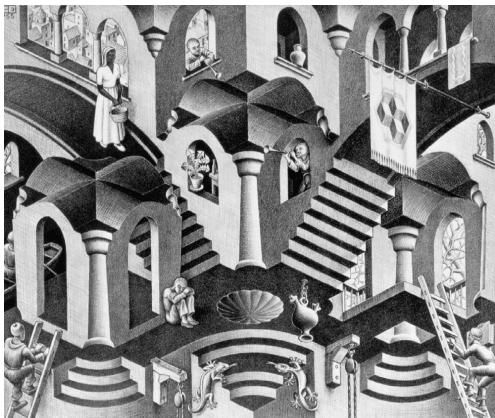


Figura 7: Esempio di figura con didascalia laterale (l'immagine riproduce la litografia *Concavo e convesso* di M. Escher)

```
\begin{SCfigure}[]
\centering
\includegraphics[width=0.5\textwidth]{ConcavoConvesso}
\caption{Esempio di figura con didascalia laterale}
\label{fig:sidecap}
\end{SCfigure}
```

6.5.2 Oggetti multipli

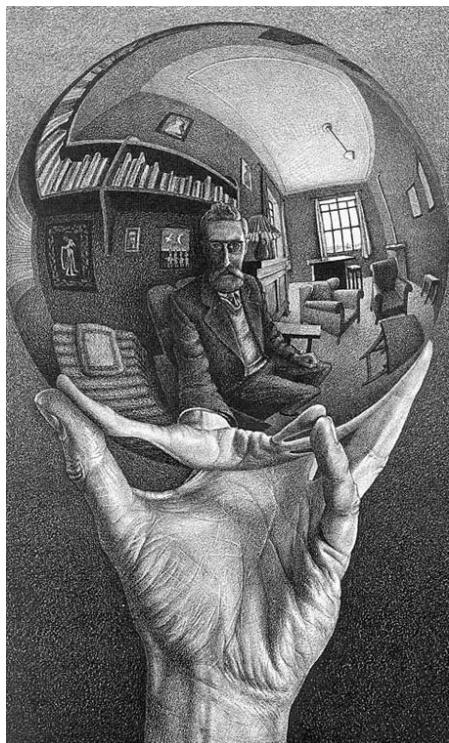
Il pacchetto `subfig` (che richiede `caption`) gestisce con `\subfloat` più (sotto)-figure (come la figura 8 nella pagina successiva) o (sotto)tabelle in un unico ambiente mobile e ne regola le didascalie molto finemente.

La figura 8 nella pagina seguente 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}} \\
\subfloat[][\emph{Cascata}] {
    \includegraphics[width=.45\textwidth]{Cascata}} \quad
\subfloat[][\emph{Salita e discesa}] {
    \includegraphics[width=.45\textwidth]{SalitaDiscesa}}
\caption{Esempio di figura composta da più sottofigure}
\label{fig:subfig}
\end{figure}
```

Si noti che:

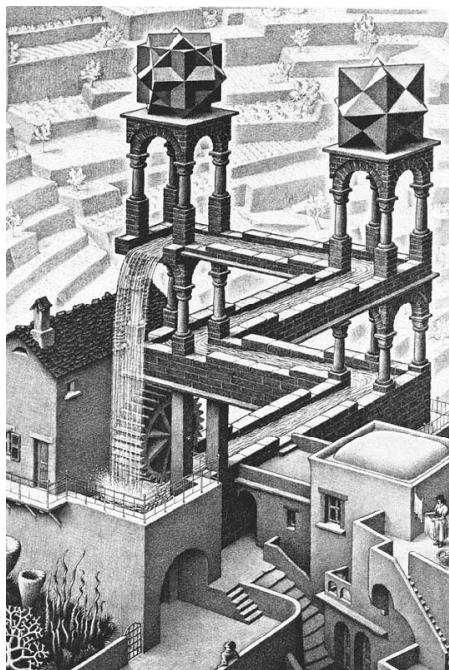
- nel primo argomento facoltativo di `\subfloat`, se usato, si mette la didascalia breve da mandare nel relativo indice (`\listoffigures` o `\listoftables`), purché si impostino adeguatamente i contatori `lofdepth` e `lotdepth` (si veda la documentazione di `subfig`);
- nel secondo ci va la didascalia 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.



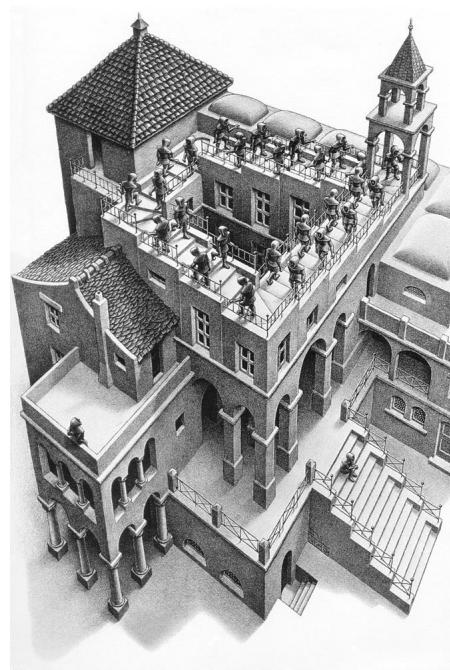
(a) *Mano con sfera riflettente*



(b) *Belvedere*



(c) *Cascata*



(d) *Salita e discesa*

Figura 8: Esempio di figura composta da più sottofigure (le immagini riproducono alcune litografie di M. Escher)

6.5.3 Oggetti immersi nel testo

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.

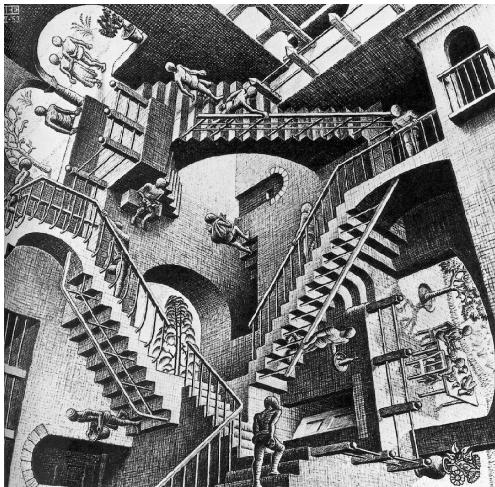


Figura 9: Esempio di figura immersa nel testo (l'immagine riproduce la litografia *Relatività* di M. C. Escher)

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 praticamen-

te *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 9 è stata ottenuta con un codice del tipo:

Talvolta si può voler “avvolgere” un oggetto con del testo.

```
\begin{wrapfloat}{figure}{I}{0pt}
\includegraphics[width=0.5\textwidth]{Relativo}
\caption{Esempio di figura immersa nel testo}
\end{wrapfloat}
```

Per avere un buon risultato, possono servire alcuni aggiustamenti manuali.

Come si vede nell'esempio precedente, l'ambiente `wrapfloat` richiede tre argomenti obbligatori:

```
\begin{wrapfloat}{<oggetto>}{{<collocazione>}}{<larghezza>}
```

Dove:

- $\langle \text{oggetto} \rangle$ indica il tipo di *oggetto* da includere (`figure` o `table`, da non confondere con gli ambienti omonimi);

Tabella 46: 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>)

- $\langle\text{collocazione}\rangle$, che dice a `LATeX` dove mettere l’oggetto sulla pagina, accetta *una sola* delle otto opzioni della tabella 46, in minuscolo o in maiuscolo a seconda che si voglia mettere l’oggetto “esattamente qui nel testo” o si voglia creare un oggetto mobile, rispettivamente;
- $\langle\text{larghezza}\rangle$ 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 L^AT_EX 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 L^AT_EX 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}{<etichetta più lunga>}
\item[<etichetta personalizzata>]{<chiave di citazione>}
...
\end{thebibliography}
```

dove:

- *<etichetta più lunga>* può essere un numero (di solito 9 se la bibliografia comprende meno di dieci opere, 99 se almeno dieci ma meno di cento, eccetera) oppure del testo (nel caso di etichette personalizzate: si scrive allora l'*etichetta più lunga*);
- `\item` va premesso a ogni riferimento bibliografico;
- *<etichetta personalizzata>* è un'*etichetta personalizzata* che *eventualmente* comparirà nella bibliografia e nelle citazioni al posto del numero predefinito;

- *<chiave di citazione>* è l’etichetta *univoca* per citare la fonte nel documento (si consiglia di usare la sintassi *<autore>:<titolo>* analoga a quella di *\label*).

Lo si vede all’opera nell’esempio seguente:

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

\bibitem{mori:tesi}
Mori, Lapo 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 \LaTeXe”, *\Ars*~(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` 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 successivo).

7.2.1 Biber

Da qualche anno il nuovo motore bibliografico predefinito da `biblatex` è Biber. `BIBTEX` 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 `LATEX` presentati nel paragrafo 2.1.1 a pagina 11, tuttavia, sono ancora preimpostati per lavorare con `BIBTEX`. In attesa che la situazione si stabilizzi, si danno di seguito le semplici istruzioni per passare a Biber:

TEXSTUDIO Si segua il percorso `opzioni → Configure TeXstudio...` e nella riga `BibTeX` si sostituisca `biber` a `bibtex`.

TEXSHOP Si segua il percorso `TeXShop → Preferenze... → Motore` e nella riga `BibTeX Engine` si sostituisca `biber` a `bibtex`.

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       = {2017},
    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 <oggetto>:<parola chiave> (dove <parola chiave> 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

<nome del campo> = {<contenuto del campo>} ,

e, come si vede, va terminato con la virgola, *anche se è l'ultimo*, pena un errore.

Per indicare a \LaTeX quale o quali database usare per comporre la bibliografia si dà nel preambolo il comando

\addbibresource{<nome del database>.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 aggiungervi informazioni.

I tipi di record e i relativi 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 135).

@article Articolo apparso in una rivista o in un giornale.

Campi obbligatori: *author*, *title*, *journalttitle*, *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 135).

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 135).

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 (ovvero `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 ([Becari, 2012]).

Uso: documenti in cui è importante indicare questi 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].

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.*

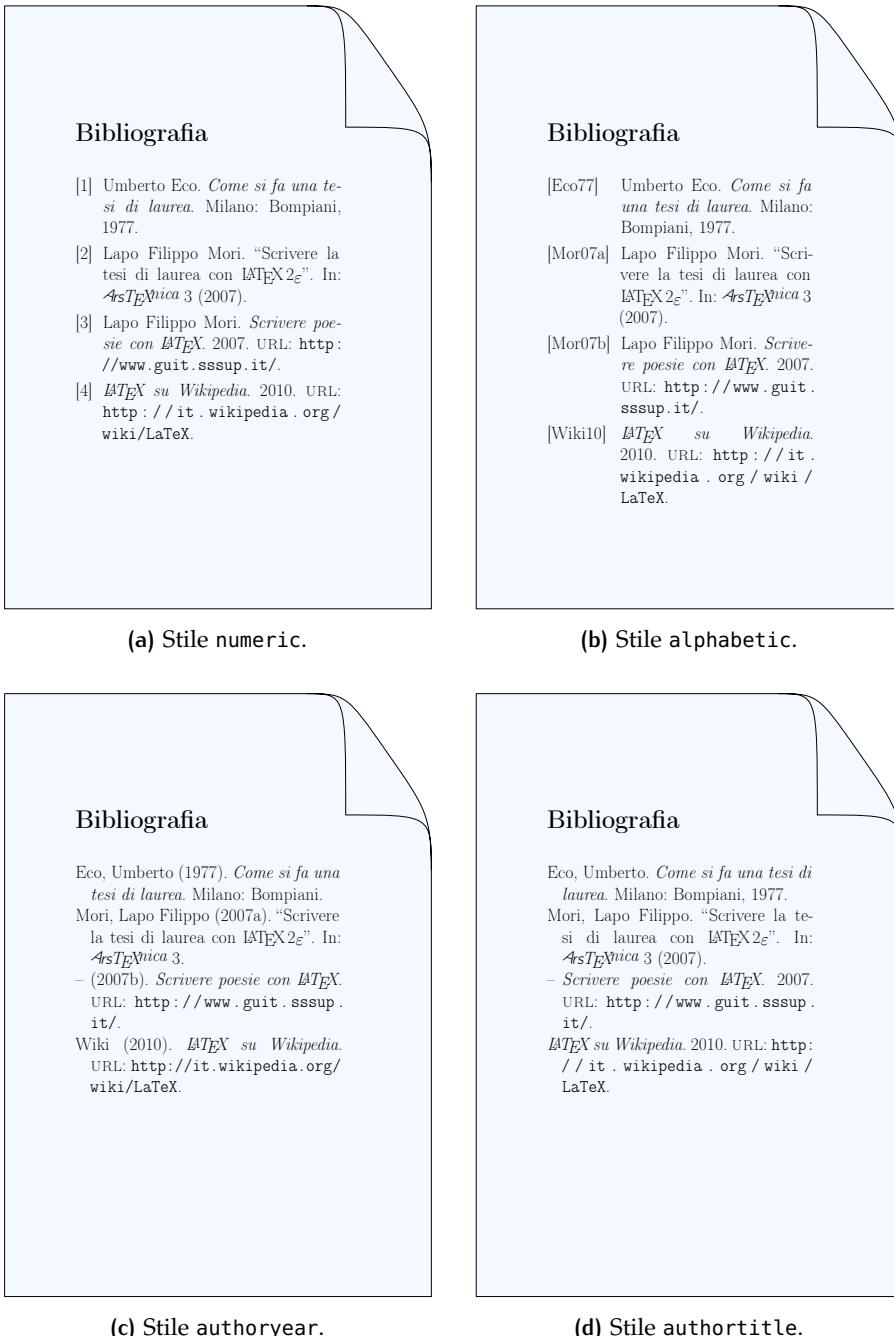
Per l’elenco completo degli schemi di citazione di biblatex si rimanda alla documentazione del pacchetto.

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}
```

**Figura 10:** Esempi di stili bibliografici

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 123.

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 11a a pagina 137);
- 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.

¹ [Eco, 1977](#).

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 `\citet` 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. Per mandarne il relativo titolo nell'indice generale si usa l'istruzione:

<code>\printbibliography[heading=bibintoc]</code>	
---	--

che funziona con le classi `article`, `report` e `book`.

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 (almeno) 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 virgolette, 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 127 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).

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{lamport:latex,
    author      = {Leslie Lamport},
    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 `miabliografia.bib` e si consideri il seguente codice:

```
\documentclass{article}
\usepackage[italian]{babel}
\usepackage[autostyle,italian=guillemets]{csquotes}
\usepackage[style=alphabetic]{biblatex}
```

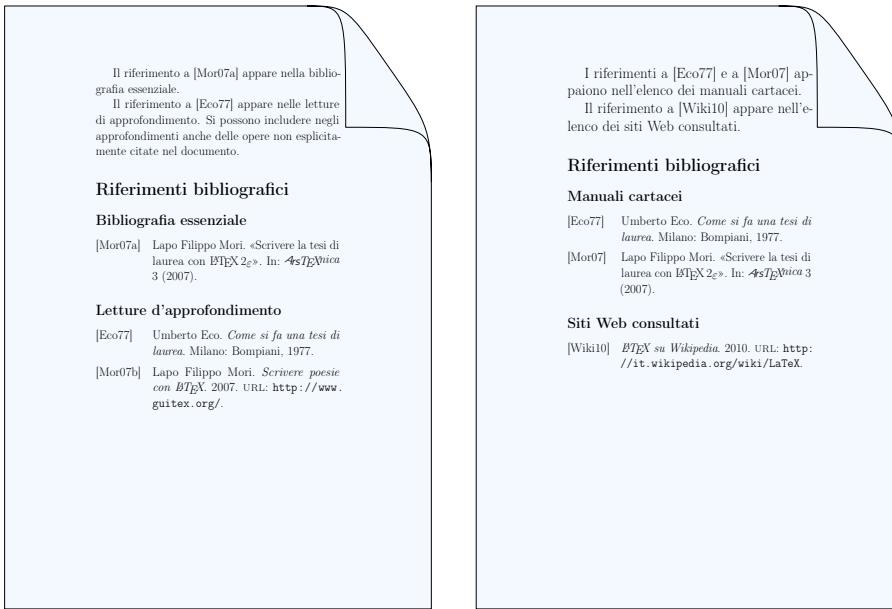


Figura 11: Bibliografie suddivise

```
\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 11a nella pagina precedente.

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 è `<tipo>`.

Il risultato è mostrato nella figura 11b nella pagina precedente.

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 L^AT_EX è capace di gestirlo automaticamente e con grande efficienza, come si mostra in questo capitolo.

8.1 INDICI SEMPLICI

Per creare l'indice analitico con L^AT_EX bisogna innanzitutto eseguire due operazioni preliminari:

1. caricare il pacchetto `imakeidx` per abilitare il programma alla composizione dell'indice;
2. dare *nel preambolo* il comando `\makeindex` (con eventuali opzioni, spiegate di seguito), 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 47 nella pagina seguente, nel cui argomento in teoria si può scrivere ciò che si vuole.

Per produrre l'indice analitico basta dare `\printindex` *immediatamente prima* di `\end{document}` e compilare (una volta) con L^AT_EX.

Per esempio, il codice:

```
\documentclass{article}
...
\usepackage{imakeidx}
\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
\end{document}
```

crea un indice analitico con le voci *arte* e *interi espressioni come questa*.

Per produrre la sezione dell'indice analitico e *mandarne il titolo nell'indice generale* basta dare l'opzione `intoc` a `\makeindex`:

```
\makeindex[intoc]
```

Tabella 47: Sintassi del comando \index

Tipo di voce e codice	Risultato
Primaria \index{Artisti}	Artisti, 2
Sottovoce \index{Artisti!Escher}	Artisti, 2 Escher, 3
Voce in altro stile \index{Gaudì@\textit{Gaudì}}	Gaudì, 5
Pagina in altro stile \index{Klimt textbf{}}	Klimt, 7
Rimando \index{Picasso see{Cubismo}}	Picasso, <i>si veda</i> Cubismo
Intervallo di pagine \index{Mirò {} ... \index{Mirò })}	Mirò, 11–13

8.2 INDICI COMPLESSI

Con le semplici istruzioni presentate nel paragrafo precedente si può comporre un indice analitico più che dignitoso. Il pacchetto `imakeidx`, tuttavia, permette di gestire e personalizzare molto finemente l'indice analitico, e di produrre anche indici multipli. Di seguito si descrivono le principali funzioni di `imakeidx` e si rinvia alla sua documentazione per gli approfondimenti.

8.2.1 Comporre l'indice

Il comando

`\makeindex[⟨opzioni⟩]`

ha un argomento facoltativo costituito da un elenco di *⟨opzioni⟩*, separate da virgole. Un'opzione può essere un'unica parola o una voce *⟨chiave⟩=⟨valore⟩*. Le principali chiavi disponibili sono le seguenti.

name È il nome dell'indice; serve solo negli indici multipli.

Esempio: `name=persone`.

title È il titolo dell'indice. Il suo valore predefinito è `\indexname` (*Indice analitico*, in italiano).

Esempio: `title=Indice dei nomi`.

options È un elenco di opzioni di ordinamento e formato. Permette di usare uno stile personale, contenuto in un file *⟨stile⟩.ist* (si veda il paragrafo 8.2.6 a pagina 142).

Esempio: `options=stile.ist`.

intoc Manda il titolo dell'indice analitico nell'indice generale.

columns Indica il numero di colonne dell'indice.

Esempio: `columns=3`.

columnsep Indica la distanza tra le colonne dell'indice.

Esempio: `columnsep=15pt.`

columnseprule Stampa una riga tra le colonne dell'indice.

8.2.2 Indicizzare le voci

Il comando

`\index[<nome>]{<voce>}`

inserisce la `<voce>` nell'indice chiamato `<nome>`, se si usano indici multipli. Se non si specifica alcun `<nome>`, la `<voce>` va nell'indice predefinito.

8.2.3 Inserire una premessa nell'indice

Il comando

`\indexprologue[<spazio>]{<testo>}`

da dare subito prima di `\printindex`, si usa per scrivere del `<testo>` fra il titolo dell'indice e le voci. Lo `<spazio>` imposta la distanza verticale fra il `<testo>` e le voci.

8.2.4 Stampare l'indice

Il comando

`\printindex[<nome>]`

stampà l'indice chiamato `<nome>`. Se non si specifica alcuna opzione, stampa l'indice predefinito.

8.2.5 Un esempio

Un esempio chiarirà le idee. Il codice:

```
\documentclass{article}
...
\makeindex[title=Indice dei concetti]
\makeindex[name=persone,title=Indice dei nomi,columns=3]

\begin{document}
La teoria della relatività\index{relatività} è stata formulata da
Albert Einstein\index{Einstein}{Albert}.

\printindex

\indexprologue{\small Questo indice contiene nomi di persone famose.}
\printindex[persone]
\end{document}
```

crea due indici analitici, il primo dei quali si intitola *Indice dei concetti* e contiene la voce *relatività*, mentre il secondo si intitola *Indice dei nomi*, contiene la voce *Einstein, Albert*, è disposto su tre colonne e ha una premessa.

8.2.6 Personalizzare l'indice analitico

L'indice analitico che si ottiene con le impostazioni predefinite non è molto elegante. Si può ricorrere a un apposito file, `stile.ist`, così definito:

```
headings_flag 1
heading_prefix "\\goodbreak\\textsc{"
heading_suffix "}\\"par\"nobreak\\vskip\\smallskipamount\\nobreak"
symhead_positive "Simboli"
symhead_negative "simboli"
numhead_positive "Numeri"
numhead_negative "numeri"
```

Una volta registrato e posizionato `stile.ist` nella cartella di lavoro, per produrre l'indice analitico personalizzato basta specificare

```
\makeindex[options=stile.ist]
```

e compilare con L^AT_EX.

9

PERSONALIZZAZIONI

Prima o poi, capita a tutti di aver bisogno di “cose” non contemplate da L^AT_EX standard: di solito si tratta di comandi e ambienti ad hoc o personalizzazioni di quelli esistenti. In questo capitolo s’impara come farlo.

9.1 COMANDI E AMBIENTI PERSONALI

9.1.1 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.

L^AT_EX 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 `\pianta`, 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 `\pianta` 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 L^AT_EX 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:

- $\langle nome \rangle$ è il nome che si dà al nuovo comando.
- $\langle numero di argomenti \rangle$ 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.
- $\langle definizione \rangle$ sono le istruzioni che specificano ciò che si vuole che il nuovo comando “faccia”. Gli argomenti eventualmente assegnati al comando s’indicano nella $\langle definizione \rangle$ 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 20):

Questo lavoro s'intitola \arte.

Questo lavoro s'intitola *L'arte di scrivere con L^AT_EX*.

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*:

```
\newcommand{\pianta}[1]{\textit{#1}}
```

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:

\pianta{Rosa canina}

Rosa canina

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].

9.1.2 Nuovi ambienti

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

```
\newenvironment{<nome>}[<numero di argomenti>]%
  {<comandi di apertura>}{<comandi di chiusura>}
```

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

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

si potrà usare il nuovo ambiente `itaitemize` così:

<pre>\begin{itaitemize} \item Un elenco con voci\texttt{dots} \item \texttt{dots} automaticamente in corsivo. \end{itaitemize}</pre>	<ul style="list-style-type: none"> • <i>Un elenco con voci...</i> • <i>...automaticamente in corsivo.</i>
--	---

Per ridefinire ambienti già esistenti, si usa il comando `\renewenvironment` (analogo a `\renewcommand`), che ha la stessa sintassi di `\newenvironment`.

9.2 PERSONALIZZARE IL TESTO

9.2.1 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 **MAIUSCOLETT**TO 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 12 e 13 nella pagina successiva mostrano le quattro categorie di font appena esaminate.

Di regola, L^AT_EX 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.

L^AT_EX 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 157, dovrebbero bastare per la gran parte delle esigenze.

In linea generale, in un documento servono diverse famiglie di font:



Figura 12: Famiglie di font a larghezza variabile



Figura 13: Famiglie di font a larghezza fissa

- 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 serie di font 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 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 su [FONTS](#).

9.2.2 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 18 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. Nell'esempio si è usata la dichiarazione \em per mettere in evidenza il testo straniero, secondo le consuetudini italiane. In modo analogo si può ridefinire l'ambiente otherlanguage* (o otherlanguage, se serve) da usare come al solito.

9.3 SPECIALITÀ

Questo paragrafo descrive alcune delle numerose possibilità offerte da L^AT_EX per personalizzare il proprio documento.

9.3.1 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}{<numero>}
<testo multicolonna>
\end{multicols}
```

dove:

- $\langle numero \rangle$ è il numero delle colonne (da 2 a 10) in cui s'intende dividere il testo;
- $\langle testo\ multicollonna \rangle$ si spiega da sé; si ottiene un testo su più colonne, bilanciate automaticamente e sezionate con i comandi consueti. L'ambiente `multicols`, che si può usare per un intero documento o solo per alcune sezioni di esso, non accetta note a margine né oggetti mobili che non siano a tutta pagina, ciò che lo rende adatto 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).

9.3.2 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 con 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 soltan-

Tabella 48: Parole fisse italiane di babel

Comando	Voce	Comando	Voce
\abstractname	Sommario	\indexname	Indice analitico
\alsoname	vedi anche	\listfigurename	Elenco delle figure
\appendixname	Appendice	\listtablename	Elenco delle tabelle
\bibname	Bibliografia	\pagename	Pag.
\ccname	e p. c.	\partname	Parte
\chaptername	Capitolo	\prefacename	Prefazione
\contentsname	Indice	\proofname	Dimostrazione
\enclname	Allegati	\refname	Riferimenti bibliografici
\figurename	Figura	\seename	vedi
\glossaryname	Glossario	\tablename	Tabella
\headtoname	Per		

to 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{\langle colore \rangle}{\langle testo \rangle}
```

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** si usa il comando

```
\colorbox{\langle colore \rangle}{\langle testo \rangle}
```

tenendo presente che il riquadro modificherà l'interlinea del capoverso e non verrà spezzato su più righe.

9.3.3 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\langle lingua \rangle{\langle testo \rangle}
```

che comanda a L^AT_EX 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 48 elenca i comandi di queste voci con la relativa traduzione italiana.

9.3.4 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 L^AT_EX.

9.3.5 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.

`tocloft` Permette di cambiare la resa tipografica degli indici (generale, delle tabelle e delle figure) e di definirne di nuovi.

`fancyhdr` Permette di personalizzare ogni aspetto di testatine e piedi.

`enumitem` Permette di personalizzare ogni aspetto degli ambienti standard per gli elenchi e di definirne di nuovi.

10 | REVISIONE FINALE

Come si è detto più volte, anche se L^AT_EX 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.

10.1 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 L^AT_EX 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 L^AT_EX non riesce a comporre bene può influenzare la divisione in pagine del documento, per cui sistemarlo *potrebbe* risolvere au-

tomaticamente 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 sicuri di “curarle” tutte, si possono evidenziare dando subito prima dell’inizio del documento il comando

`\overfullrule=<lunghezza>`

che accanto a ciascuna riga che sporge dal margine stampa un ■ (più o meno grande a seconda del valore assegnato a *<lunghezza>*, 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 \- 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 rifor- mulando il testo; si eviti che un capoverso termini con una sola parola molto breve (o, peggio, con una sola sillaba) racchiudendola in `\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 cer- te sequenze: tra una lettera corsiva con tratti discendenti come la *f* e una parentesi tonda immediatamente prima, per esempio.

10.2 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 ri- ferimento 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”;

- 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 L^AT_EX ha messo in una pagina con del testo.

Come fare? I primi due casi si risolvono semplicemente arretrando il codice dell'oggetto di qualche capoverso, mentre il quarto dandogli la preferenza p, che L^AT_EX 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. A volte può risolvere la situazione \clearpage, spiegato nel paragrafo 10.2 a pagina 155.

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 certosina lo conferma la terminologia tedesca: una riga orfana si chiama *Schusterjunge* ("apprendista ciabattino"), mentre una riga vedova *Hurenkind* ("figlio di p***"). L^AT_EX 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 L^AT_EX 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.

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 31 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;
- 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 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 L^AT_EX 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 presente che L^AT_EX interrompe automaticamente la pagina nel punto ritenuto migliore: i comandi descritti ne modificano il comportamento predefinito e possono risolvere situazioni problematiche, ma vanno usati con 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], 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 L^AT_EX. 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.

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 preferibilmente solo 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.

Gli accenti vanno apposti *sopra* la vocale, minuscola o maiuscola che sia: il paragrafo 3.2.1 a pagina 18 mostra come impostare i pacchetti che permettono di interpretare correttamente le lettere accentate italiane. Al contrario di quanto accade con le vecchie macchine per scrivere o con altri elaboratori di testo, L^AT_EX permette di accentare senza problemi *anche le lettere maiuscole*, che spesso vengono erroneamente apostrofatte (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'*, 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.

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 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'interpunkzione 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 interpunkzione, seguono immediatamente la parola che li precede e sono separati con uno spazio da quella che li segue... in questo modo. Con L^AT_EX, 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.

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 L^AT_EX. 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».

Il *tratto*, preceduto e separato da uno spazio bianco, di solito suddivide elementi come titoli e didascalie, come in «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 L^AT_EX.

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 **MAIUSCOLETT**. 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 L^AT_EX. Lo stile “evidenziato” (o “enfatizzato”) 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: «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* è 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 utilizzato 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. Talvolta si usa per i titoli di certe sezioni.

A.4 TRATTAMENTO DEL TESTO

Ci limiteremo a prendere 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 18).

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 <code>\$1821\$</code> non è primo. \\ Napoleone morì nel <code>1821</code> .	Il numero <code>1821</code> non è primo. \\ Napoleone morì nel <code>1821</code> .
--	--

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).

I numeri minuscoli presentano tratti ascendenti e discendenti per integrarsi al meglio con i caratteri minuscoli del font corrente, e non sono indicati in tutti gli altri casi. Di solito per ottenerli non serve fare nulla, perché sono predefiniti nel font scelto: si consiglia in ogni caso di fare una prova per verificarlo.

Spaziare le cifre

La corretta scrittura dei numeri interi *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` del pacchetto `siunitx`. Si osservi il risultato tipografico dei numeri negli esempi seguenti:

<code>\$1500000\$ \\</code>	1500000
<code>\$1\,,500\,,000\$ \\</code>	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 per cento 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 e 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 5.10 a pagina 88).

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 `GIT`, `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* 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: `GUIt` (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 (come *sig.* o *prof.*). 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).

ACRONIMI

A_MS	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' A_MS 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 _E X Archive Network
	La “rete di archivi completi di T _E X” è, nel Web, il luogo di riferimento da cui scaricare software e materiale su T _E X e L _A T _E X.
G_IT	Gruppo Utilizzatori Italiani di T _E X e L _A T _E X
	È un’associazione senza fini di lucro con lo scopo di aumentare la diffusione di T _E X e L _A T _E X 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 163 Paesi del mondo.
JPEG	Joint Photographic Experts Group
	È il più usato standard di compressione delle immagini fotografiche. Esegue una compressione con perdita di informazioni (cioè di tipo <i>lossy</i>). 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. Esegue una compressione senza perdita di informazioni (cioè di tipo <i>lossless</i>). Con il JPG , è il formato standard delle immagini bitmap da inserire in un documento da comporre con L_AT_EX .

TUG	TeX User Group
	L'espressione (“gruppo di utenti di TeX ”) indica le associazioni di persone accomunate dalla passione per TeX e LATeX 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' <i>host</i> 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 (<i>Unicode Transformation Format-8 bit</i>) 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 wysiwyg .
WYSIWYM	What You See Is What You Mean
	L'acronimo (“ciò che vedi è ciò che intendi”) è stato coniato espressamente per LATeX in contrapposizione a wysiwyg (nel senso di “programma di videoscrittura caratterizzato da una composizione sincrona”).
	La caratteristica che più differenzia LATeX dagli altri elaboratori di testo è il fatto che per realizzare un documento con questo programma bisogna agire <i>in tempi diversi</i> 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 LATeX , che lo compone e lo impagina per produrre il documento finito.

ELENCO DEI SITI INTERNET

CTAN

Sito ufficiale di CTAN.

<http://www.ctan.org/>

FONTS

Catalogo dei font per L^AT_EX.

<http://www.tug.dk/FontCatalogue/>

GUIT

Sito ufficiale del G_UI_T.

<http://www.guitex.org/>

MACTEX

Pagina Web di MacT_EX.

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

TEXCATALOGUE

Catalogo L^AT_EX online.

<http://texcatalogue.ctan.org/>

TEXLIVE

Pagina Web di T_EX Live.

<http://www.tug.org/texlive/>

TEXSHOP

Pagina Web di T_EXShop.

<http://pages.uoregon.edu/koch/texshop/>

TEXSTUDIO

Pagina Web di T_EXstudio.

<http://www.texstudio.org/>

BIBLIOGRAFIA

Beccari, Claudio

- 2016 *Introduzione all'arte della composizione tipografica con L^AT_EX*, <http://www.guitex.org/home/images/doc/guidaguit-b5.pdf>.

Beccari, Claudio e Tommaso Gordini

- 2015 *L'arte di scrivere in diverse lingue con X^AT_EX*, <http://www.guitex.org/home/images/doc/ArteLingue.pdf>.
- 2016 *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>.

Bringhurst, Robert

- 1992 *The Elements of Typographic Style*, Hartley & Marks, Vancouver; trad. it. *Gli Elementi dello Stile Tipografico*, Sylvestre Bonnard, Milano 2001.

Cevolani, Gustavo

- 2006 «Norme tipografiche per l'italiano in L^AT_EX», *ArstExnica*, 1, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica01.pdf.

De Marco, Agostino

- 2014 *Scrivere la tesi di laurea in L^AT_EX*, <http://www.guitex.org/home/images/doc/GuideGuIT/IntroTesi.pdf>.

Eco, Umberto

- 1977 *Come si fa una tesi di laurea. Le materie umanistiche*, Bompiani, Milano.

Fairbairns, Robin

- 2014 *The UK T_EX FAQ*, <http://texdoc.net/texmf-dist/doc/generic/FAQ-en/newfaq.pdf>.

Goossens, Michel, Frank Mittelbach e Johannes Braams

- 2004 *The L^AT_EX Companion*, Addison-Wesley, Reading (Massachusetts).

Gregorio, Enrico

- 2009 *Appunti di programmazione in L^AT_EX e T_EX*, <http://profs.sci.univr.it/~gregorio/introtex.pdf>.
- 2010 *L^AT_EX. Breve guida ai pacchetti di uso più comune*, <http://profs.sci.univr.it/~gregorio/breveguida.pdf>.
- 2013 *Installare T_EXLive 2012 su Ubuntu*, <http://profs.sci.univr.it/~gregorio/texlive-YEAR-ubuntu.pdf>.

Guiggiani, Massimo e Lapo Filippo Mori

- 2008 «Come non maltrattare le formule matematiche», *ArstExnica*, 5, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica05.pdf.

Knuth, Donald Ervin

- 1973 *Computer Programming as an Art*, Addison-Wesley, Reading (Massachusetts), vol. 3.

- Knuth, Donald Ervin
 1984 *The TeXbook*, Addison-Wesley, Reading (Massachusetts).
- Lamport, Leslie
 1994 *LaTeX. A Document Preparation System*, Addison-Wesley, Reading (Massachusetts).
- Mittelbach, Frank, Gianluca Pignalberi e Dave Walden
 2007 «Intervista a Frank Mittelbach», *ArstEXnica*, 3, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica03.pdf.
- Mori, Lapo Filippo
 2006 «Tabelle su $\text{\LaTeX}_2\epsilon$: pacchetti e metodi da utilizzare», *ArstEXnica*, 2, http://www.guit.sssup.it/arstexnica/download_ars/arstexnic02.pdf.
 2007 «Scrivere la tesi di laurea con $\text{\LaTeX}_2\epsilon$ », *ArstEXnica*, 3, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica03.pdf.
- Oetiker, Tobias, Hubert Partl, Irene Hyna e Elisabeth Schlegl
 2011 *The Not So Short Introduction to LaTeX2ε*, <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>.
- Pakin, Scott
 2015 *The Comprehensive LaTeX Symbol List*, <http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>.
- Pantieri, Lorenzo
 2017 *LaTeXpedia*, http://www.lorenzopantieri.net/LaTeX_files/LaTeXpedia.pdf.
- Sabatini, Francesco e Vittorio Coletti
 1997 *il Sabatini Coletti. Dizionario della Lingua Italiana*, Giunti, Firenze.
- Serianni, Luca
 1989 *Grammatica italiana. Italiano comune e lingua letteraria*, con la collaborazione di Alberto Castelvecchi, UTET, Torino.

INDICE ANALITICO

COMANDI SPECIALI

\!, 74, 75
\#, 21, 22
\\$, 21, 22
\%, 21, 22
\&, 21, 22
\', 54
\(), 65
\), 65
\,, 69, 74, 75, 152, 160, 162
\-, 50, 152
\., 54
\=, 54
\@, 51
\[, 66, 92
\], 24, 48, 49, 56, 61, 62,
98, 107, 113,
114, 151
\{, 21, 23
\}, 21, 23
\], 66, 92
\^, 54
_, 21, 22
\', 54
\~, 21, 54

A

\AA, 54
\aa, 54
\abstractname, 149
\addcontentsline, 37, 40
\addtocategory, 137
\AE, 54
\ae, 54
\aleph, 80
\alpha, 71
\alsoname, 149
Ambiente
 abstract, 32
 align, 39, 80–82
 align*, 82
 aligned, 82
 array, 91, 92, 94, 99,
100
 Bmatrix, 78
 bmatrix, 78
 cases, 78, 82
 center, 92, 95
 comment, 27
 description, 59
 document, 44
 doublespace, 31
 enumerate, 39, 59
 eqnarray, 66
 eqnarray*, 66
 equation, 39, 66, 82
 equation*, 66

figure, 39, 94, 118,
148
figure*, 148
gather, 39, 80, 81
gather*, 82
gathered, 82
itaitemize, 145
itemize, 59
landscape, 114
matrix, 78
multicols, 148
multiline, 39, 80
multiline*, 81
onehalfspace, 31
otherlanguage, 136
otherlanguage, 20,
147
otherlanguage*,
147
pmatrix, 78
proof, 87
quotation, 61
quote, 61
quoting, 62
SCfigure, 118
SCfigure*, 118
SCTable, 118
SCTable*, 118
sidewaysfigure,
112
sidewaystable, 112
singlespace, 31
smallmatrix, 79
split, 39, 80, 81
subequations, 82
table, 39, 93, 111,
118, 148
table*, 148
tabular, 78, 82, 91,
92, 94, 99, 100,
111, 112
tabularx, 101
thebibliography,
32, 123–125
titlepage, 57
verbatim, 63
verse, 62
Vmatrix, 78
vmatrix, 78
wrapfloat, 121, 122
\AmS, 53
\and, 25, 56
\angle, 80
\ap, 54
\appendix, 22, 34
\appendixname, 149
\approx, 81

\arccos, 76
\arcsin, 76
\arctan, 76
\arg, 76
\arraybackslash, 110
\Ars, 21, 53
\ast, 80
\asymp, 81
\author, 25

B

\b, 54
\backmatter, 33
\bar, 71, 72
\beta, 71
\bfseries, 55
\bibitem, 123
\bibname, 125, 149
\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
\bigdot, 80
\bigoplus, 80
\bigotimes, 80
\Bigr, 77
\bigr, 77
\bigskip, 49
\binom, 77
\blacksquare, 80
\bmod, 76
\boldsymbol, 78, 84
\bottomrule, 99
\Bra, 78
\Braket, 78
\bullet, 80

C

\c, 54
\cap, 70
\caption, 94
\captionsetup, 95
\ccname, 149
\cdot, 80
\centering, 95, 109
\chapter, 32, 38, 39, 58
\chaptername, 149
\check, 72

\chi, 71
 \circ, 80
 \cite, 124, 134
 \citeauthor, 135
 \citet, 135
 \citet, 135
 \citeyear, 135
Classe
 article, 28, 29, 34, 35, 125, 135
 beamer, 28
 book, 28, 29, 33, 35, 36, 125, 135, 138
 KOMA-Script, 28
 letter, 28
 memoir, 28
 report, 28, 29, 35, 125, 135, 138
 softes, 28, 147
 toptesi, 28
 \cleardoublepage, 125, 155
 \clearpage, 125, 153, 155
 \cmidrule, 99
 \colon, 73
 \color, 149
 \columnwidth, 148
 \complement, 70
 \cong, 81
Contatore
 secnumdepth, 37, 38
 tocdepth, 37
 toctumdepth, 38
 \contentsname, 149
 \cos, 75, 76
 \cosh, 76
 \cot, 76
 \coth, 76
 \csc, 76
 \cup, 70

D

\d, 54
 \dagger, 80
 \dashv, 80
 \date, 56
 \ddagger, 80
 \ddot, 72
 \ddots, 79
 \DeclareMathOperator, 76
 \defbibheading, 138
 \deg, 76
 \Delta, 71
 \delta, 71
 \det, 76
 \frac, 85, 107
 \mathbb{D}, 54
 \mathfrak{d}, 54
 \diamond, 80
 \dim, 76
 \displaystyle, 84

\div, 80
 \mathbb{D}\mathbb{J}, 54
 \mathfrak{d}\mathfrak{j}, 54
 \documentclass, 21, 22, 24, 28
 \dot, 72
 \dots, 21, 26, 52, 53, 73, 159
 \double spacing, 31
 \Downarrow, 73
 \downarrow, 73

E

\ell, 80
 \mathfrak{m}, 55, 147
 \emph, 21, 22, 55
 \emptyset, 70
 \enclname, 149
 \endfirsthead, 113
 \endfoot, 113
 \endhead, 113
 \endlastfoot, 113
 \epsilon, 71
 \eqref, 67
 \equiv, 81
 \eta, 71
 \euro, 53
 \exists, 74
 \exp, 76

F

\figurename, 149
 \footcite, 134
 \footnote, 39, 111
 \footnotetext, 56
 \forall, 74
 \foreignlanguage, 147
 \frenchspacing, 51
 \frontmatter, 33, 35
 \fullcite, 134

G

\Gamma, 71
 \gamma, 71
 \mathbb{g}b, 144
 \gcd, 76
 \ge, 81
 \gets, 73
 \mathfrak{g}g, 81
 \glossaryname, 149
 \graphicspath, 45
 \mathbb{G}\mathbb{U}\mathbb{T}, 53
 \mathbb{G}\mathbb{U}\mathbb{T}^*, 53

H

\mathbb{H}, 54
 \hat, 72
 \hbar, 80
 \headtoname, 149
 \hline, 98
 \hom, 76
 \hookleftarrow, 73
 \hookrightarrow, 73

\href, 40
 \mathfrak{h}\mathfrak{s}\mathfrak{l}\mathfrak{a}\mathfrak{sh}, 80
 \hspace, 107
 \mathbb{H}\mathbb{U}\mathbb{G}, 56
 \huge, 56
 \hypersetup, 39
 \hyphenation, 50

I

\dotsint, 69
 \mathbb{I}\mathbb{I}\mathbb{F}, 74
 \iiint, 69
 \iiint, 69
 \mathbb{I}\mathbb{I}\mathbb{I}\mathbb{N}, 80
 \imath, 80
 \mathbb{I}\mathbb{M}\mathbb{A}\mathbb{H}, 80
 \implies, 74
 \mathbb{I}\mathbb{N}, 70
 \include, 45
 \includegraphics, 91, 92, 116, 122
 \includeonly, 45
 \index, 140
 \indexname, 140, 149
 \inf, 76
 \infty, 69
 \input, 44, 45
 \int, 69
 \iota, 71
 \item, 22, 59
 \itshape, 22, 55

J

\mathbb{J}\mathbb{M}\mathbb{A}\mathbb{H}, 80

K

\mathbb{k}, 54
 \kappa, 71
 \ker, 76
 \mathbb{K}\mathbb{e}\mathbb{t}, 78

L

\mathbb{L}, 54
 \mathfrak{l}, 54
 \label, 38, 67, 87, 94, 119, 124
 \Lambda, 71
 \lambda, 71
 \mathbb{L}\mathbb{A}\mathbb{N}\mathbb{D}, 74
 \langle, 89, 159
 \mathbb{L}\mathbb{A}\mathbb{R}\mathbb{G}, 56
 \Large, 56
 \large, 56
 \mathbb{L}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}, 17, 21, 22, 53
 \mathbb{L}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}\mathbb{E}, 53
 \le, 81
 \left, 77
 \mathbb{L}\mathbb{E}\mathbb{F}\mathbb{A}\mathbb{R}\mathbb{W}\mathbb{O}, 73
 \leftarrow, 73
 \mathbb{L}\mathbb{E}\mathbb{F}\mathbb{L}\mathbb{E}\mathbb{F}\mathbb{A}\mathbb{R}\mathbb{W}\mathbb{O}, 73
 \mathbb{L}\mathbb{E}\mathbb{F}\mathbb{R}\mathbb{I}\mathbb{G}\mathbb{H}\mathbb{A}\mathbb{R}\mathbb{W}\mathbb{O}, 73
 \leftarrow, 73
 \mathbb{L}\mathbb{E}\mathbb{F}\mathbb{R}\mathbb{I}\mathbb{G}\mathbb{H}\mathbb{A}\mathbb{R}\mathbb{W}\mathbb{O}, 73
 \lg, 76

\lim, 76
 \liminf, 76
 \limsup, 76
 \linespread, 22
 \listfigurename, 149
 \listoffigures, 32, 119
 \listoftables, 32, 119
 \listtablename, 149
 \ll, 81
 \ln, 76
 \log, 75, 76
 \Longleftarrow, 73
 \longleftarrow, 73
 \Longleftrightarrow, 73
 \longleftarrowright, 73
 \longmapsto, 73
 \Longrightarrow, 73
 \longrightarrow, 73
 \lvert, 74
 \lVert, 72
 \lvert, 72

M

\mail, 40
 \mainmatter, 33
 \makeindex, 139
 \maketitle, 25, 57
 \MakeUppercase, 36
 \mapsto, 73
 \markboth, 35, 36
 \markright, 35
 \mathbb, 70, 83
 \mathbf, 78, 83
 \mathcal, 83
 \mathfrak, 83
 \mathit, 83
 \mathrm, 83
 \mathscr, 83
 \mathsf, 83
 \mathtt, 83
 \max, 76
 \mbox, 152
 \medskip, 49
 \MF, 53
 \mid, 72, 81
 \midrule, 98
 \MiKTeX, 53
 \min, 76
 \models, 74
 \MP, 53
 \mp, 67
 \mu, 71
 \multicolumn, 22, 100

N

\nabla, 80
 \ne, 81
 \nearrow, 73
 \neg, 74
 \newcolumntype, 108
 \newcommand, 143, 144
 \newenvironment, 144, 145

\newline, 49, 101
 \newpage, 155
 \newtheorem, 39, 85
 \newtheorem*, 85
 \newtheoremstyle, 86
 \exists, 74
 \ni, 70
 \nocite, 134
 \noindent, 49
 \normalsize, 55, 56
 \notag, 82
 \notin, 70
 \nu, 71
 \num, 162
 \nwarrow, 73

O

\o, 54
 \oe, 54
 \oint, 69
 \Omega, 71
 \omega, 71
 \omissis, 53, 159
 \onehalfspacing, 31
 \oplus, 80
 \otimes, 80
 \overbrace, 77
 \overline, 71
 \overrightarrow, 71
 \owns, 70

P

Pacchetto
 acronym, 64
 afterpage, 36
 amsmath, 39, 53, 65, 76, 77, 80–82, 100, 107
 amssymb, 65
 amsthm, 85, 86
 appendix, 34
 array, 101, 108, 110
 ArsClassica, 2, 28, 147
 babel, 19, 20, 42, 54, 125, 135, 149, 158
 biblatex, 32, 123, 125–132, 134–136
 bmpsize, 115
 bookmarks, 40
 booktabs, 91, 98
 braket, 78
 caption, 91, 95, 96, 119, 121
 chemfig, 89
 chemformula, 89
 ClassicThesis, 9

ClassicThesis, 57, 147
 crop, 31
 csquotes, 62, 125
 dblfloatfix, 148
 dtklogos, 53
 emptypage, 36
 etoolbox, 149
 eurosym, 53
 float, 97, 148
 fontenc, 18, 42
 footmisc, 58
 frontespizio, 57
 geometry, 30
 glossaries, 64
 graphicx, 91, 114, 116, 118
 guil, 53
 hologo, 53
 hyperref, 37, 39, 40, 125
 imakeidx, 139, 140
 indentfirst, 49
 inputenc, 18, 42
 LayAureo, 30, 42
 listings, 64
 longtable, 112
 makeidx, 32
 mathrsfs, 83
 mathtools, 72
 mflogo, 53
 microtype, 49
 minitoc, 36
 mparhack, 58
 multicol, 147
 natbib, 135
 pdfscape, 114
 pgfplots, 115
 quoting, 62
 ragged2e, 110
 rotating, 112
 setspace, 31
 shorttoc, 36
 showidx, 155
 showkeys, 39
 sidecap, 39, 118
 siunitx, 88, 103, 105, 162, 163
 steroid, 41
 subfig, 39, 119
 supertabular, 112
 tabularx, 100–102
 textcomp, 89
 TikZ, 115
 url, 40
 varioref, 39
 verse, 63
 wrapfig, 121
 wrapfloat, 39
 xcolor, 148
 xtab, 112

\pagebreak, 153, 155
 \pagename, 149
 \pageref, 38

\par, 48, 55
 \paragraph, 32, 33, 38, 39
 \parallel, 72, 81
 \parencite, 134
 \part, 32, 33, 38, 39
 \partial, 80
 \partname, 149
 \pdfbookmark, 22
 \ped, 54
 \perp, 81
 \phantomsection, 37, 125
 \Phi, 71
 \phi, 71
 \Pi, 71
 \pi, 71
 \pianta, 143, 144
 \pm, 67
 \pmod, 76
 \Pr, 76
 \prec, 81
 \preceq, 81
 \prefacename, 149
 \printbibliography, 32,
 134, 135, 138
 \printindex, 32, 139, 141
 \prod, 68
 \proofname, 149
 \proto, 81
 \Psi, 71
 \psi, 71

Q

\qedhere, 88
 \quad, 43, 68, 74, 75, 107
 \quad, 43, 74, 75, 107

R

\r, 54
 \raggedbottom, 31
 \RaggedLeft, 110
 \raggedleft, 109
 \RaggedRight, 110
 \raggedright, 109
 \rangle, 89, 159
 \Re, 76, 80
 \ref, 38
 \refname, 125, 149
 \renewcommand, 144, 145
 \renewenvironment, 145
 \rho, 71
 \right, 77
 \rightarrow, 73
 \rightarrow, 73
 \rightarrowrightarrows, 73
 \rmfamily, 55
 \rVert, 72
 \rvert, 72

S

\scriptscriptstyle, 84
 \scriptsize, 56
 \scriptstyle, 84
 \scshape, 55
 \searrow, 73
 \sec, 76

\section, 25, 32, 33, 38,
 39, 58
 \section*, 138
 \seename, 149
 \Set, 78
 \setminus, 70, 72
 \sfamily, 55
 \sgn, 76
 \Sigma, 71
 \sigma, 71
 \sim, 81
 \simeq, 81
 \sin, 75, 76
 \singlespacing, 31
 \sinh, 76
 \sisetup, 103
 \slshape, 55
 \small, 22, 56
 \smallskip, 49
 \sqrt, 68
 \square, 80
 \subfloat, 22, 119
 \subparagraph, 32, 38, 39
 \subsection, 32, 38, 39
 \subsection*, 138
 \subset, 70
 \subseteq, 70
 \subsubsection, 32, 38,
 39
 \succ, 81
 \succeq, 81
 \sum, 68
 \sup, 76
 \supercite, 134
 \supset, 70
 \supseteq, 70
 \surd, 80
 \swallow, 73

T

\t, 54
 \tablename, 149
 \tableofcontents, 25, 32,
 94
 \tabularnewline, 109
 \tan, 76
 \tanh, 76
 \tau, 71
 \TeX, 21, 53
 \text, 67, 82, 100
 \textbackslash, 24
 \textbf, 55, 101, 143
 \textcite, 134
 \textheight, 117
 \textit, 22, 55, 59, 143
 \textrm, 55
 \textsc, 55
 \textsf, 55
 \textsl, 55
 \textstyle, 84
 \textsuperscript, 54
 \texttt, 55
 \textup, 84
 \textwidth, 31, 92, 101,
 117, 148

\tfrac, 85
 \TH, 54
 \th, 54
 \thanks, 56
 \theoremstyle, 86
 \Theta, 71
 \theta, 71
 \tilde, 72
 \tiny, 56
 \title, 25
 \to, 73
 \today, 21, 22
 \toprule, 98
 \triangle, 80
 \triangleleft, 80
 \triangleright, 80
 \ttfamily, 55

U

\u, 54
 \underbrace, 77
 \underline, 71
 \underset, 71
 \Uparrow, 73
 \uparrow, 73
 \Updownarrow, 73
 \updownarrow, 73
 \Upsilon, 71
 \upsilon, 71
 \url, 40
 \usepackage, 42

V

\v, 54
 \varepsilon, 71
 \varphi, 71
 \varpi, 70, 71
 \varrho, 71
 \varsigma, 70, 71
 \vartheta, 71
 \vdash, 80
 \dots, 79
 \vec, 71, 72, 78
 \vee, 80
 \verb, 63
 \Vert, 72
 \vert, 72
 \vref, 39
 \vspace, 49
 \vspace*, 49

W

\wedge, 80
 \widehat, 72
 \widetilde, 72
 \wp, 80

X

\Xi, 71
 \xi, 71
 \leftarrow, 73
 \rightarrow, 73

Z

\zeta, 71