

# Primi passi in L<sup>A</sup>T<sub>E</sub>X

## 1 Prime notizie

L<sup>A</sup>T<sub>E</sub>X è un linguaggio di “markup” usato per la preparazione di testi basato sul programma di composizione tipografica T<sub>E</sub>X. T<sub>E</sub>X venne creato alla fine degli anni '70 dal matematico/informatico Donald Knuth. L<sup>A</sup>T<sub>E</sub>X venne creato nel 1985 dal matematico/informatico Leslie Lamport.

Al contrario dei “word processor”, quali ad esempio *Microsoft Word*, *WordPerfect*, *Works*, che sono basati sul paradigma WYSIWYG (What You See Is What You Get, cioè ciò che vedi è quello che ottieni), con L<sup>A</sup>T<sub>E</sub>X si scrive un testo preoccupandosi essenzialmente del contenuto (della struttura) e non della forma. Il testo del documento, detto sorgente, conterrà anche delle istruzioni (direttive o comandi di L<sup>A</sup>T<sub>E</sub>X): per ottenere l'output finale è necessario che tale sorgente sia poi compilato.

Questo approccio viene anche definito WYSIWYM (What You See Is What You Mean, cioè ciò che vedi è quello che intendi): con L<sup>A</sup>T<sub>E</sub>X l'autore inizialmente può occuparsi delle convenzioni da usare, ma una volta fissate queste, si può concentrare soltanto sul contenuto del testo. L'impaginazione, l'indice (generale e analitico), l'inserimento delle figure e delle tabelle sarà semi-automaticamente curato da L<sup>A</sup>T<sub>E</sub>X.

Il file sorgente va scritto con un editor di testi. Successivamente il file va compilato e si produce un file (di estensione `.dvi`) che può essere usato per visualizzare il risultato sullo schermo del calcolatore o stamparlo. L<sup>A</sup>T<sub>E</sub>X è un prodotto non commerciale, ma di pubblico dominio. Vi sono poi molti ambienti integrati che forniscono delle interfacce più o meno accattivanti per elaborare testi in L<sup>A</sup>T<sub>E</sub>X.

## 2 I mattoni

Nel file di input (file sorgente) di un documento L<sup>A</sup>T<sub>E</sub>X si usano *solo* i seguenti caratteri:

- le 26 lettere dell'alfabeto (maiuscole e minuscole);

- le cifre 0...9;
- i seguenti 16 caratteri:

. : ; , ? ! ‘ ’ ( ) [ ] - / \* @

- i seguenti 10 caratteri speciali (cioè che per  $\text{\LaTeX}$  hanno un significato speciale):

# \$ % & ~ \_ \ { }

- i 5 caratteri:

+ = | < >

che sono prevalentemente utilizzati nelle formule matematiche;

- lo spazio e l' "a capo".

### 3 Le prime costruzioni...

$\text{\LaTeX}$  prevede che un documento sia strutturato in *unità logiche*: molti comandi di  $\text{\LaTeX}$  descrivono quello che si chiama la struttura logica del documento, cioè il documento stesso, i capitoli, le sezioni, le sottosezioni, le tabelle, le single frasi, le parole e così via fino ad arrivare alle singole lettere.  $\text{\LaTeX}$  è concepito in modo che l'utilizzatore sia concentrato prevalentemente sul testo e non sulla struttura logica, della quale si preoccupa il programma. Ad esempio ci sono comandi per definire i capitoli (in un libro), le sezioni, le sottosezioni, ecc. ed è poi  $\text{\LaTeX}$  che decide (salvo un intervento dell'utilizzatore) sull'aspetto grafico del testo: il font e le dimensioni da usare per i titoli dei capitoli, delle sezioni, delle sottosezioni e la loro numerazione, come dividere in sillabe le parole, dove spezzare le righe in modo da avere la giustificazione del margine destro...

La fine di una parola è specificata con uno o più spazi. La fine di un paragrafo è ottenuto con una o più righe bianche. Per le sottosezioni, sezioni e capitoli vedremo i comandi specifici.

## 4 . . . e le prime pignolerie

Ci sono 3 modi di fare “il trattino” in  $\text{\LaTeX}$ : si usa il carattere - una, due o tre volte.

- Si usa - per scrivere due parole unite da trattino, come ad esempio: “sub-normale” (che si scrive `sub-normale`);
- Si usa -- per indicare un “range” numerico: ad esempio “pagg. 34-47” (si scrive “`pagg. 34--47`”);
- si usa --- per i trattini parentetici (si dice così?), come, tanto per fare un esempio—almeno spero—questo, che si scrive:  
`...esempio---almeno spero---questo.`

Le “virgolette” si fanno o con: ‘ seguito da: ’ o con: ‘ ‘ seguito da: ’’.

Esempi: ‘ciao’ e “ciao” (che si scrivono, rispettivamente, ‘ciao’ e ‘ ‘ciao’ ’).

Talvolta è spiacevole il modo in cui  $\text{\LaTeX}$  va “a capo”. Ad esempio la frase “... a pagina 3 abbiamo visto che ...” sarebbe spiacevole se fosse scritta:

```
“...   ...   ...   a pagina
3 abbiamo visto che ...”
```

In questo caso si usa il carattere speciale: ~ al posto dello spazio, quindi si scrive: “a pagina~3 abbiamo visto che”. L’effetto è:

```
“...   ...   ...   a pagina 3
abbiamo visto che ...”
```

quindi “~” si comporta esattamente come uno spazio ( $\square$ ), ma dice a  $\text{\LaTeX}$  di non usare quello spazio per spezzare la riga.

## 5 Simboli speciali e primi comandi

I simboli \$ & % # \_ { } hanno un significato speciale in  $\text{\LaTeX}$ , se quindi devono essere scritti in un testo, vanno preceduti dal simbolo \. Ad esempio: per scrivere 100\$ si scrive: `100\$`. Altri caratteri che hanno un significato speciale sono: ~ ^ \. Per ottenere questi si usa, ad esempio, il comando `\verb` che sarà spiegato in seguito.

Ecco ora alcuni comandi (detti anche “direttive”) che generano del testo e, a fianco, il loro effetto:

Comando	Effetto (descrizione)
<code>\today</code>	27 maggio 2008 (data odierna)
<code>\LaTeX</code>	L <sup>A</sup> T <sub>E</sub> X (logo di L <sup>A</sup> T <sub>E</sub> X)
<code>\TeX</code>	T <sub>E</sub> X (logo di T <sub>E</sub> X)
<code>\ldots</code>	... (puntini di sospensione)
<code>\_</code>	(uno spazio)

I comandi di L<sup>A</sup>T<sub>E</sub>X finora visti consistono di una barra obliqua (`\`), seguita da uno o più caratteri. Molti altri comandi consistono invece di una barra obliqua, seguita da una stringa di caratteri e uno o più argomenti (racchiusi tra parentesi graffe). Ad esempio:

Comando	Descrizione/effetto
<code>\textit{argomento}</code>	<i>Scrivo l'argomento in italico</i>
<code>\textsl{argomento}</code>	<i>Scrivo l'argomento inclinato</i>
<code>\textsc{argomento}</code>	SCRIVE L'ARGOMENTO IN "MAIUSCOLETTO"
<code>\textbf{argomento}</code>	<b>Scrivo l'argomento in "grossetto"</b>
<code>\texttt{argomento}</code>	Scrivo l'argomento nei caratteri della macchina da scrivere
<code>\emph{argomento}</code>	<i>Scrivo l'argomento "enfaticizzato" (solitamente enfaticizzato = italico)</i>

## 5.1 Significato dei simboli speciali

Il simbolo `%` ha due utilizzi: può essere usato per inserire un commento (tutti i caratteri che seguono `%` fino alla fine della riga sono ignorati nella compilazione), oppure può essere usato per finire una riga senza produrre uno spazio nell'output. Ad esempio, scrivendo:

```
... precipitevoli%
ssimevolmente...
```

si ottiene "precipitevolissimevolmente" (e non "precipitevoli ssimevolmente", come invece succederebbe se non avessimo usato il `%`).

Il simbolo speciale `$` si usa per scrivere formule matematiche, (di più in seguito).

Il simbolo `&` serve per scrivere array e tabelle (di più in seguito).

Il simbolo `#` si usa nell'ambito della definizione di nuovi comandi (per indicare la presenza di un argomento—di più in seguito).

Le parentesi graffe `{ }` si usano per racchiudere opportunamente del testo.

## 6 Formule - primi passi

Quando in un testo ci sono formule matematiche,  $\text{\LaTeX}$  prevede particolari opzioni. Le formule che compaiono nel mezzo di una frase vanno inserite entro due simboli di dollaro ( $\$$ ). Ad esempio:

Sia  $x$  un numero reale e sia  $y = x + 1$

va scritta:

Sia  $x$  un numero reale e sia  $y = x + 1$

Al posto dei due segni di  $\$$  si possono usare, in maniera del tutto analoga, rispettivamente,

$\left($  e  $\right)$

oppure, rispettivamente,

$\begin{math}$  e  $\end{math}$ .

## 7 Il documento

L'unità logica principale è il documento stesso. La prima istruzione di  $\text{\LaTeX}$  è quindi quella che specifica il tipo di documento che si vuol produrre. Un documento  $\text{\LaTeX}$  comincia sempre con il comando:  $\backslash\text{documentclass}\{\dots\}$  dove l'argomento può essere `article` o `report` o `book` o altro ancora. Ad esempio questo testo comincia con:

```
 $\backslash\text{documentclass}\{\text{article}\}$ 
```

Per essere precisi, l'istruzione completa è:

```
 $\backslash\text{documentclass}[12\text{pt}, \text{a4paper}]\{\text{article}\}$ 
```

poiché si possono usare varie opzioni. Eccone alcune: (il simbolo `|` in questo caso indica un'alternativa)

```
10\text{pt}|11\text{pt}|12\text{pt}
```

(definisce le dimensioni di default del documento),

```
\text{letterpaper}|\text{legalpaper}|\text{executivepaper}|\text{a4paper}|\text{a5paper}|\text{b5paper}
```

(definisce la formattazione dell'output delle dimensioni appropriate—esempio: `a4paper` è  $210\text{mm} \times 297\text{mm}$ ).

```
\text{onecolumn}|\text{twocolumn}
```

(stabilisce se il testo va scritto in una o due colonne).

Alcuni esempi possono essere:

```
 $\backslash\text{documentclass}[\text{twocolumn}, \text{a4paper}, 12\text{pt}]\{\text{article}\}$ 
```

```
 $\backslash\text{documentclass}[10\text{pt}, \text{b5paper}]\{\text{book}\}$ 
```

```
 $\backslash\text{documentclass}[11\text{pt}, \text{twocolumn}]\{\text{report}\}$ 
```

Il testo di ogni documento deve cominciare poi con  $\backslash\text{begin}\{\text{document}\}$  e finire con  $\backslash\text{end}\{\text{document}\}$ . Tra la direttiva iniziale  $\backslash\text{documentclass}$  e

`\begin{document}` ci possono essere varie istruzioni che costituiscono quello che si chiama il *preambolo*. Nel preambolo non si scrive il testo del documento, ma si forniscono alcune istruzioni di carattere generale che saranno funzionali nel corpo del documento. Nel preambolo si inseriscono ad esempio comandi per aver accesso ad alcuni “pacchetti” della libreria: si usa l’istruzione `\usepackage[opzioni]{nome pacchetto}`. Quindi per caricare il pacchetto `graphics` che aggiunge funzionalità grafiche a  $\text{\LaTeX}$  si usa `\usepackage{graphics}`; per caricare un pacchetto che istruisce  $\text{\LaTeX}$  a dividere in sillabe le parole secondo le regole della lingua italiana (e non inglese) si carica il pacchetto `babel` con l’istruzione: `\usepackage[italian]{babel}`

Solitamente un documento ha un titolo e un autore. Sia il titolo, sia l’autore, sono informazioni che possono essere utilizzate più di una volta (ad esempio il titolo potrebbe essere ripetuto in testa ad ogni pagina). Il modo “pulito” previsto da  $\text{\LaTeX}$  per inserire autore e titolo è di inserire nel preambolo le direttive: `\title{...}` e `\author{...}`. Successivamente, *dopo* l’istruzione `\begin{document}` si inserisce il comando `\maketitle`. In questo modo viene stampato il titolo con l’autore e in aggiunta la data (del giorno in cui viene compilato il file). Per inserire una data decisa da noi, basta aggiungere nel preambolo la direttiva `\date{...}` dove nell’argomento, che può essere vuoto, si inserisce l’informazione voluta.

Un tipico esempio delle prime istruzioni di un documento  $\text{\LaTeX}$  potrebbe quindi essere quello mostrato in figura 1.

## 8 Capitoli, sezioni, sottosezioni...

Un testo è composto da lettere che formano parole; le parole formano frasi; le frasi sono strutturate in paragrafi e i paragrafi costituiscono sezioni che a loro volta fanno parte di capitoli. Questa struttura logica di un testo si rende in  $\text{\LaTeX}$  con vari comandi. Una parola è delimitata da uno spazio bianco o un comando di “a capo”; un paragrafo è delimitato da una riga vuota; una sezione—nelle classi standard di  $\text{\LaTeX}$  che sono `article`, `report` e `book`—è definita dal comando `\section{titolo}` (una sezione può a sua volta contenere sottosezioni (comando `\subsection{titolo}`), che possono contenere sottosottosezioni (comando `\subsubsection{titolo}`), e così via); un capitolo è definito dal comando `\chapter{titolo}` (non esiste nella classe `article`). Si noti che  $\text{\LaTeX}$  fornisce automaticamente una numerazione alle “unità logiche” ora descritte.

```
\documentclass[a4paper, 12pt]{book}
%%%% inizio del preambolo

\usepackage[italian]{babel} % per la div. in sillabe
\usepackage{graphics} % aggiunge potenzialita' grafiche

\title{Dialogo dei Massimi Sistemi}
\author{Galileo Galilei}
\date{10 settembre 1624}
...
...
%%%% fine preambolo

\begin{document}

\maketitle
...
...
\end{document}
```

Figura 1: Il tipico inizio un documento  $\LaTeX$

## 9 Materiale in esposizione

Certe volte è necessario inserire nel testo una citazione, o scrivere un elenco, o una poesia, inserire cioè del materiale che va evidenziato in qualche modo particolare.  $\text{\LaTeX}$  prevede alcune specifiche istruzioni. Ad esempio una citazione può essere racchiusa tra un `\begin{quote}` e un `\end{quote}`. L'effetto è quello di scrivere il testo con entrambi i margini spostati.

In questo esempio si è incontrato un tipo di costrutto di  $\text{\LaTeX}$  che è denominato *environment* (ambiente) e che è caratterizzato dalla sintassi:

```
\begin{nome} ... \end{nome}
```

dove *nome* è l'ambiente specifico. Già si sono incontrati due altri esempi di ambiente: l'ambiente `document` e l'ambiente `math`.

Altri ambienti utili sono quelli relativi alle *liste*. Ci sono tre possibili casi: `enumerate`, `itemize` e `description`. I componenti di una lista vanno preceduti dall'istruzione `\item`.

Ecco alcuni esempi (a sinistra il risultato, a destra l'input). Il primo esempio si riferisce all'ambiente `enumerate`:

Testo normale.

1. Primo elemento della lista;
2. Secondo elemento della lista;
3. Terzo elemento della lista.

Riprende il testo normale.

---

Testo normale.

```
\begin{enumerate}
\item Primo elemento
della lista;
\item Secondo elemento
della lista;
\item Terzo elemento
della lista.
\end{enumerate}
Riprende il testo normale.
```

Il secondo esempio si riferisce all'ambiente `itemize`:

Testo normale.

- Primo elemento della lista;
- Secondo elemento della lista;
- Terzo elemento della lista.

Riprende il testo normale.

---

Testo normale.

```
\begin{itemize}
\item Primo elemento
della lista;
\item Secondo elemento
della lista;
\item Terzo elemento
della lista.
\end{itemize}
Riprende il testo normale.
```

Infine, il terzo esempio si riferisce all'ambiente `description`. Si noti che in questo caso il comando `\item` viene usato con un argomento.

Testo normale.	Testo normale.
<b>uno:</b> Primo elemento della lista;	<code>\begin{description}</code> <code>\item[uno:]</code> Primo elemento della lista;
<b>due:</b> Secondo elemento della lista;	<code>\item[due:]</code> Secondo elemento della lista;
<b>tre:</b> Terzo elemento della lista.	<code>\item[tre:]</code> Terzo elemento della lista.
	<code>\end{description}</code> Riprende il testo normale.

---

Riprende il testo normale.

Le liste possono essere anche concatenate.

Altro materiale che in un testo, soprattutto se matematico, può essere in qualche modo evidenziato, è quello relativo alle formule.  $\text{\LaTeX}$  fornisce due ambienti per evidenziare formule: `displaymath` ed `equation`. I due ambienti si comportano in modo del tutto analogo, salvo che, nel secondo, la formula scritta viene etichettata con un numero. In alternativa a `\begin{displaymath}` e a `\end{displaymath}` si possono usare, rispettivamente, i comandi: `\[` e `\]`. Se ad esempio nel file di input scriviamo (anticipando un po' di comandi che si analizzeranno in seguito):

```
\begin{displaymath}
\int e^x dx = e^x + C
\end{displaymath}
```

otteniamo la seguente formula scritta centrata:

$$\int e^x dx = e^x + C$$

se invece scriviamo:

```
\begin{equation}
\int e^x dx = e^x + C
\end{equation}
```

otteniamo lo stesso risultato, ma in più la formula viene numerata (con numeri progressivi):

$$\int e^x dx = e^x + C \tag{1}$$

## 9.1 Centrare o appoggiare al solo margine destro

Gli ambienti: `center` e `flushright` servono per scrivere del testo centrato o, rispettivamente, appoggiato a destra.

Ad esempio:

```
\begin{center}
Questo testo\\
verrà tutto centrato\\
e scritto su tre righe
\end{center}
```

L'effetto è:

Questo testo  
verrà tutto centrato  
e scritto su tre righe

(Per andare a riga nuova con un testo centrato si usa il comando: `\\`)  
Analogamente:

```
\begin{flushright}
Questo testo\\
viene scritto \\
su 5 righe\\
tutte appoggiate al margine\\
destro.
\end{flushright}
```

L'effetto è:

Questo testo  
viene scritto  
su 5 righe  
tutte appoggiate al margine  
destro.

## 9.2 Riportare del testo così come scritto in input

L'ambiente `verbatim` permette di stampare il testo così come viene scritto in input (in questo caso gli spazi e le righe bianche contano).

Esempio:

Il testo scritto qui  
viene riportato letteralmente  
nell'elaborazione di `\LaTeX`. I comandi  
di `\LaTeX` non funzionano.

Lo si vede anche da questa riga:  $\int_a^b f(x) dx$

L'input è:

```
\begin{verbatim}
Il testo scritto qui
    viene riportato letteralmente
nell'elaborazione di \LaTeX. I comandi
di \LaTeX non funzionano...
```

`\end{verbatim}` Infine, quando di devono scrivere alcuni caratteri nella riga di testo che devono essere riportati alla lettera, si usa il comando `\verb` il cui argomento non è racchiuso tra parentesi graffe, ma tra due caratteri uguali, decisi volta per volta (che possono essere un qualunque carattere che non compare nell'argomento di `verb`, tranne uno spazio, una lettera o `*`). Ad esempio `\end{document}` si può scrivere: `\verb+\end{document}+` (o, anche `\verb/\end{document}/`, oppure `\verb:\end{document}:`).