

# Scrivere Documenti con $\text{\LaTeX}$

Mario Di Raimondo

Associazione Software Libero Ragusa (SoLiRa)

linux@evening 2010



## Guide gratuite



Nadia Garbellini.

*L<sup>A</sup>T<sub>E</sub>X Facile – Guida all'Uso.*

[http://www.istitutomajorana.it/index.php?option=com\\_content&task=view&id=486&Itemid=33](http://www.istitutomajorana.it/index.php?option=com_content&task=view&id=486&Itemid=33)



Oetiker, Tobias.

*Una (mica tanto) breve introduzione a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.*

<http://www.ctan.org/tex-archive/info/lshort/italian/>

## Perché si chiama T<sub>E</sub>X?

Il nome deriva dalle prime tre lettere della parola

τεχνή (tecnica, arte)  
e  
τεχνολογία (tecnologia)

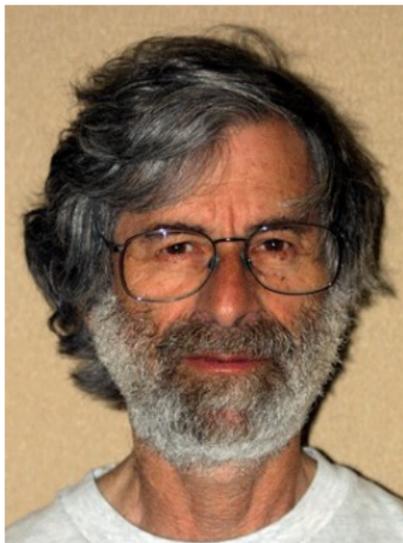
L'ultima lettera di T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X deve essere quindi letta come il  
“*ch*” di chiave

# Ecco chi ha scritto il $\text{T}_{\text{E}}\text{X}$



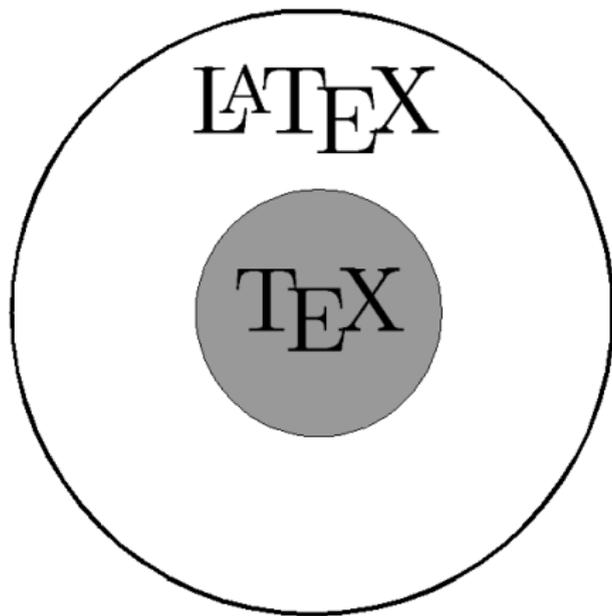
Donald E. Knuth

# Ecco chi ha sviluppato L<sup>A</sup>TEX



Leslie Lamport

# TEX è il “motore” di L<sup>A</sup>TEX



## Esistono diverse varianti di LaTeX

- **teTeX** per Unix e GNU/Linux
- **MiKTeX** per Windows
- **gwTeX** per Mac OS X
- **TeXLive**: multiplatforma, è in grado di funzionare senza essere installato
- **OzTeX**, **AmigaTeX**, ...

Tutte queste versioni differiscono tra loro solo per il sistema operativo su cui devono essere installate

## Cosa non è L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X *non* è un programma WYSIWYG  
(*what you see is what you get*)

A differenza di questo tipo di programmi **L<sup>A</sup>T<sub>E</sub>X non possiede un'interfaccia grafica** capace di visualizzare in *tempo reale* il documento pronto per la stampa

### Il concetto di compilazione

La compilazione è l'elaborazione di una serie di istruzioni, raccolte in un file di *input* (puro testo), che produce un file di *output* (per esempio un PDF).



## Cos'è L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X è simultaneamente **un programma ed un linguaggio** per la composizione tipografica, specificamente concepito per la realizzazione di documenti di elevata qualità finale.

Contrariamente a quanto si pensa la preparazione di un documento in grado di rispettare precisi canoni estetici è un lavoro assai delicato.

## Il concetto di compilazione

La compilazione è l'elaborazione di una serie di istruzioni, raccolte in un file di *input* (puro testo), che produce un file di *output* (per esempio un PDF).

Nei programmi WYSIWYG questo avviene in tempo reale. Con L<sup>A</sup>T<sub>E</sub>X invece questi due step sono tenuti separati.

### Attenzione!

D'ora in avanti ci riferiremo al file di input chiamandolo con il termine convenzionale di: **sorgente**

## Il file sorgente

Si definisce **sorgente** del documento il testo del nostro documento con all'interno tutte le istruzioni necessarie a L<sup>A</sup>T<sub>E</sub>X per formattarlo.

Questo file avrà estensione `.tex`

```
Il mio cane Ricky lo ingoia e corre tutto il giorno  
con l'ouverture di \textit{Guglielmo Tell} in  
pancia\dots
```

Il mio cane Ricky lo ingoia e corre tutto il giorno con l'ouverture di *Guglielmo Tell* in pancia...



## Gli *step* di compilazione

.tex

.tex



.dvi

## Cosa occorre

Ovviamente un compilatore L<sup>A</sup>T<sub>E</sub>X ([MikT<sub>E</sub>X](#), [teT<sub>E</sub>X](#), ecc.)

Per scrivere il file sorgente (`.tex`) è consigliabile utilizzare un *editor* di testo che aiuti a gestirne la compilazione ([T<sub>E</sub>XnicCenter](#), [WinEdt](#), [L<sup>A</sup>T<sub>E</sub>XEditor](#) (LEd), [Kile](#), [Emacs](#), [T<sub>E</sub>Xmaker](#), [VimL<sup>A</sup>T<sub>E</sub>Xsuite](#), ecc.)

Fanno anche comodo:

- visualizzatore PDF ([Acrobat Reader](#), [xpdf](#), ecc.)
- compilatore PostScript (tipicamente [GhostScript](#))
- visualizzatore PS ([gv](#), [KGhostView](#), ecc.)
- gestore della bibliografia ([bibtool](#), [BibT<sub>E</sub>Xmgr](#), ecc.)
- ...

## Ricapitolando

- si scrive il sorgente del documento (`.tex`)
- si *compila* il sorgente, ovvero dice a L<sup>A</sup>T<sub>E</sub>X di trasformare il sorgente in un documento di output (nel nostro caso un `.pdf`)
- si legge il documento prodotto con un visualizzatore per `.pdf`
- se si vuole modificare il documento bisogna modificare il sorgente e ricompilare

## La sintassi di base

- tutti i comandi cominciano sempre con un `\`
- spesso il comando è il nome inglese dell'azione
- il comando “termina” con uno spazio bianco o con un altro comando:

```
\comando <testo>
```

```
\comando\altrocomando
```

Attenzione!

L<sup>A</sup>T<sub>E</sub>X è *case sensitive*! Bisogna pertanto stare attenti a distinguere tra

**MAIUSCOLO** e **minuscolo**



## I principali tipi di comandi

Comandi semplici

```
\newpage
```

Comandi che richiedono un argomento

```
\textit{Guglielmo Tell}
```

Comandi che richiedono uno (o più) parametri

```
\vspace{2cm}
```

Alcuni comandi richiedono di specificare una o più opzioni:

```
\documentclass[12pt]{article}
```



## Caratteri riservati

Esistono poi alcuni caratteri riservati:

\$ & % # ^ \_ { } ~

che hanno un significato speciale per L<sup>A</sup>T<sub>E</sub>X e che non possono essere usati normalmente. Per poterli inserire nel documento dovranno essere tutti preceduti da un \

### Le graffe sotto Windows

Le parentesi graffe sono fondamentali e non si trovano sulle tastiere italiane, c'è un rimedio:

{ = AltGr + Maiusc + [  
} = AltGr + Maiusc + ]



## Scrivere i loghi

Ecco come si scrivono i loghi:

```
\TeX  
\LaTeX  
\LaTeXe
```

```
TeX  
LATeX  
LATeX 2ε
```

## Ambienti

Gli *ambienti* sono strutture contraddistinte da

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

Possono essere anche annidati l'uno dentro l'altro a condizione che l'ordine di chiusura sia speculare a quello di apertura

# Il modello di un documento

```
\documentclass{<classe>}
```

# Le classi base di L<sup>A</sup>T<sub>E</sub>X

```
\documentclass{<classe>}
```

- article
- report
- book
- letter
- slides
- ...
- beamer
- ...

## Il modello di un documento

```
\documentclass{<classe>
  <preambolo>

\begin{document}
  <testo del documento>
\end{document}
```

# Un esempio vale più di mille parole

`esempio_1_1.tex`

## Le opzioni di `\documentclass`

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

- 8pt ÷ 12pt
- a4paper, a5paper, ...
- titlepage
- twocolumn
- twoside
- ...

Le opzioni sono funzionali alla classe di documento prescelta

## Esempio di classe di documento

```
\documentclass[a4paper,12pt,twoside]{article}
```

Realizza un *articolo* su un foglio **A4** con carattere a **12pt** ottimizzato per la stampa **fronte/retro**.

Il bello di L<sup>A</sup>T<sub>E</sub>X

Queste impostazioni globali sono modificabili in qualsiasi momento

## Commentare il testo

Commentare il testo significa renderlo invisibile al processo di compilazione, risulta pertanto utile per escludere temporaneamente porzioni di testo o codice

```
% Prendete una persona, versatele dentro cinque o  
sei litri di birra e ne farete un ubriaco
```

```
sei litri di birra e ne farete un ubriaco
```

Attenzione!

Il commento è valido solo fino alla fine della riga!

## I file di stile

L<sup>A</sup>T<sub>E</sub>X ha una struttura modulare e prevede la possibilità di caricare delle **funzionalità aggiuntive** (*package*, pacchetti o moduli di estensione) alle funzionalità già disponibili nella dotazione di base ed indispensabili per ottenere determinate *feature*.

I pacchetti hanno estensione `.sty` e vanno richiamati all'interno del preambolo con il comando:

```
\usepackage{<nomepkg>}
```

```
\usepackage[<opzioni>]{<nomepkg>}
```

## Due esempi di pacchetti

```
\usepackage{graphicx}
```

`graphicx` è un pacchetto che permette di gestire l'inserimento delle immagini, dei colori e di rotazioni

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

`babel` permette di sillabare testi scritti in lingue diverse dall'inglese (default), attivando la sillabazione della lingua selezionata (in questo caso, la nostra: `italian`)

# Un esempio vale più di mille parole

`esempio_1_2.tex`

## L'encoding di un documento

A causa della sua vocazione multiplatforma e multilingua di L<sup>A</sup>T<sub>E</sub>X, è necessario specificare nel sorgente la codifica usata dal vostro computer per definire alcuni caratteri particolari (nel nostro specifico caso le vocali accentate). Questo sistema di codifica prende il nome di *encoding*.

Quello che utilizziamo nello standard europeo è l'**ISO-8859-15**

Attenzione!

La codifica da specificare dipende *anche* dal programma utilizzato per scrivere

## I principali *encoding* e *inputenc*

ISO-8859-1	⇒	latin1
ISO-8859-15	⇒	latin9
UTF-8	⇒	utf8, utf8x <sup>a</sup>
Codepage 1252 (Windows)	⇒	ansinew
MacRoman (Mac OS X)	⇒	applemac

---

<sup>a</sup>richiede `unicode`

Per piattaforma Windows

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

Per piattaforma \*nix

```
\usepackage[utf8x]{inputenc}
```



# Un esempio vale più di mille parole

`esempio_1_3.tex`

## Miti sfatati: meglio gli editor WYSIWYG

La cosa scomoda di L<sup>A</sup>T<sub>E</sub>X è che non vedi quello che ottieni. . .

### La verità

- con L<sup>A</sup>T<sub>E</sub>X non ci sono distrazioni, è possibile finalmente pensare solo ai contenuti
- scrivere in L<sup>A</sup>T<sub>E</sub>X aiuta a strutturare meglio il proprio lavoro, rendendolo più chiaro
- se fosse necessario è possibile comunque controllare il *layout* come (meglio) in Word

## Miti sfatati: lo posso fare con Word

Anche Word permette di definire una bibliografia dinamica, comandi di sezionamento, etc.

### La verità

- Cattive abitudini: meno dell'1% degli utenti scrive una vera sezione invece di "Sezione 1"
- L<sup>A</sup>T<sub>E</sub>X offre un controllo più profondo e vasto, è possibile anche scrivere musica o riviste di scacchi
- le macro L<sup>A</sup>T<sub>E</sub>X funzionano meglio: vogliamo fare una gara sulla gestione delle figure?

## Miti sfatati: L<sup>A</sup>T<sub>E</sub>X è difficile

### La verità

- Non ci vuole una grande fantasia per capire cosa fanno i comandi `\section` o `\footnote`
- difficile è capire perché stampando Word sposta le figure dove gli pare

Ciò che è veramente difficile è realizzare documenti disomogenei e non strutturati

# Perché strutturare

Strutturare un documento significa:

- avere le idee chiare su cosa si sta scrivendo
- organizzare i contenuti in parti, capitoli, sezioni e sottosezioni
- rendere i contenuti del documento consistenti e coerenti
- rendere partecipe il computer di cosa si desidera ottenere

## Comandi di sezionamento

```
\part{}  
\chapter{}  
\section{}  
\subsection{}
```

```
\subsubsection{}  
\paragraph{}  
\subparagraph{}
```

L<sup>A</sup>T<sub>E</sub>X si occupa automaticamente della spaziatura, stile, dimensione del titolo e dell'inserimento di questo nell'indice

## Capitolo (1)

```
\chapter{La figura di Renzo nei Promessi Sposi se  
avesse avuto un cellulare}
```

### Capitolo 1

La figura di Renzo nei Promessi Sposi se  
avesse avuto un cellulare

## Capitolo (2)

```
\chapter*{La figura di Renzo nei Promessi Sposi se  
avesse avuto un cellulare}
```

La figura di Renzo nei Promessi Sposi se  
avesse avuto un cellulare

La versione asteriscata (`\chapter*`, `\section*`, ecc.) sopprime la numerazione.

## Sezione

```
\section{La figura paradigmatica di Renzo}
```

### 1.1 La figura paradigmatica di Renzo

```
\section*{La figura paradigmatica di Renzo}
```

### La figura paradigmatica di Renzo

# Un esempio vale più di mille parole

`esempio_2_1.tex`

## Indici

L<sup>A</sup>T<sub>E</sub>X provvede in modo automatico alla generazione dell'indice sulla base della struttura da noi indicata

```
\tableofcontents  
\listoftables  
\listoffigures
```

Ognuno di questi comandi inseriti *nel corpo del documento* realizza automaticamente in quel preciso punto l'indice specifico.

### Attenzione!

Affinché venga compilato l'indice occorre compilare all'inizio *due volte* il documento (solo la prima volta)

# Titolo del documento

Per stampare il titolo dell'intero documento

- riempire i campi `\title{}`, `\author{}` e `\data{}` del template (eventualmente lasciando alcuni di essi vuoti);
- scrivere il comando `\maketitle` nel punto del testo in cui si vuole che L<sup>A</sup>T<sub>E</sub>X generi il titolo.

## Titolo del documento

```
\title{Le confessioni di un formaggio mostruoso}  
\author{Hans Metterling}  
\data{\today}  
  
\maketitle
```

# Le confessioni di un formaggio mostruoso

Hans Metterling

6 marzo 2010

# Un esempio vale più di mille parole

`esempio_2_2.tex`

## Elenchi puntati

```
\begin{itemize}
  \item Pippo
  \item Paperino
  \item Paperoga
\end{itemize}
```

- Pippo
- Paperino
- Paperoga

## Elenchi puntati personalizzati

```
\begin{itemize}
  \item[-] Pippo
  \item[*] Paperino
  \item[$\surd$] Paperoga
\end{itemize}
```

- Pippo
- \* Paperino
- ✓ Paperoga

## Elenchi numerati

```
\begin{enumerate}  
  \item Pippo  
  \item Paperino  
  \item Paperoga  
\end{enumerate}
```

1. Pippo
2. Paperino
3. Paperoga

## Descrizioni

```
\begin{description}  
  \item[Pippo] è sfortunato  
  \item[Paperino] è molto sfortunato  
  \item[Paperoga] è il più sfortunato di tutti  
\end{description}
```

Pippo è sfortunato  
Paperino è molto sfortunato  
Paperoga è il più sfortunato di tutti

# Un esempio vale più di mille parole

`esempio_2_4.tex`

## Nota a pié di pagina

```
[\dots] sono persone simpatiche con cui scambiare  
due chiacchiere durante la sosta\footnote{0 meglio  
lo erano. La Commissione per il Controllo Fluviale  
sembra essersi trasformata in un sindacato per il  
collocamento degli idioti.}
```

[...] sono persone simpatiche con cui scambiare due chiacchiere durante la sosta<sup>a</sup>.

---

<sup>a</sup>O meglio lo erano. La Commissione per il Controllo Fluviale sembra essersi trasformata in un sindacato per il collocamento degli idioti.

## Uno spazio e due a capo

L<sup>A</sup>T<sub>E</sub>X:

- non distingue uno spazio da molti spazi
- non dà importanza al fatto che una riga sia interrotta da un 'a capo': per dire di chiudere un paragrafo occorre lasciare una linea vuota
- interrompe una riga *senza* cominciare un nuovo paragrafo (comportamento generalmente da evitare) in presenza `\\`

## Singolo ‘a capo’

Un solo ‘a capo’ non produce alcun effetto così come pure diversi spazi bianchi:

```
[\dots] riuscì a sapere che      Lambertini viveva a  
Sasso Marconi in una villa signorile.  
Ma recatosi sul posto trovò solo una cuccia da cani  
alta due metri in stile tirolese [\dots]
```

```
[...] riuscì a sapere che Lambertini viveva a Sasso Marconi  
in una villa signorile. Ma recatosi sul posto trovò solo una  
cuccia da cani alta due metri in stile tirolese [...]
```

## Nuovo paragrafo

Per cominciare un nuovo paragrafo bisogna lasciare una riga vuota

```
[\dots] riuscì a sapere che Lambertini viveva a Sasso  
Marconi in una villa signorile.
```

```
Ma recatosi sul posto trovò solo una cuccia da cani  
alta due metri in stile tirolese [\dots]
```

[...] riuscì a sapere che Lambertini viveva a Sasso Marconi  
in una villa signorile.

Ma recatosi sul posto trovò solo una cuccia da cani alta due  
metri in stile tirolese [...]

## Allineamento di *default*

L<sup>A</sup>T<sub>E</sub>X giustifica di *default* il testo nel documento, mantenendo la stessa distanza (variabile) fra le parole e sillabandole correttamente se non riesce a “impaginare” le parole sulla riga.

L’algoritmo è infinitamente più efficiente di quello di Word

## Centrata del testo

```
\begin{center}  
  I sette gnomi di Zurigo  
\end{center}
```

I sette gnomi di Zurigo

I *comandi* prendono effetto fino alla fine del gruppo in cui sono racchiusi; tale gruppo può essere formato sia dalle parentesi graffe (“{”, “}”) sia da un ambiente.

Nel caso si voglia un comando globale si può usare `\centering`

```
\centering
```

```
I sette gnomi di Zurigo
```

I sette gnomi di Zurigo

Attenzione!

Se non è chiuso in nessun gruppo, il comando prende effetto fino alla fine del documento!

## Allineamento a destra e sinistra

```
\begin{flushright}
```

La favola della fine del mondo

```
\end{flushright}
```

La favola della fine del mondo

E la dichiarazione corrispondente è `\raggedleft`

Analogamente per l'allineamento a sinistra si usa `flushleft` e `\raggedright`

# Un esempio vale più di mille parole

`esempio_2_5.tex`

## Cosa sono?

I *riferimenti incrociati* permettono di richiamare il numero di una nota, di una sezione, o di una figura o tabella o il numero di pagina di un particolare elemento che si desidera citare nel testo. In  $\text{\LaTeX}$  questi riferimenti vengono gestiti in modo automatico

### Il bello di $\text{\LaTeX}$

Il pacchetto `hyperref` trasforma i riferimenti incrociati in link, così da trasformare il documento in *ipertesto*. Anche l'indice viene inoltre trasformato in una serie di link.

# Etichettare

Nel testo del documento posso inserire delle *label* con il comando

Applico a questa slide una label `\label{<nome>}`

## Numero dell'elemento

Queste *label* possono essere richiamate in altre parti del documento con il comando:

La label si trova alla slide numero `\ref{<nome>}`.

La label si trova alla slide numero 41.

## Pagina dell'elemento

Queste label possono essere richiamate in altre parti del documento con il comando:

La label si trova alla pagina numero  
`\pageref{<nome>}`.

La label si trova alla pagina numero 61.

# Un esempio vale più di mille parole

`esempio_2_7.tex`

## Uso dell'enfasi

Il testo enfattizzato si usa per nomi propri e titoli citati, nonché per enfattizzare il testo:

Ti accorgerai che è il `\emph{tuo}` re a rischiare di essere messo sotto scacco.

Ti accorgerai che è il *tuo* re a rischiare di essere messo sotto scacco.

## Uso del corsivo

Il corsivo (italico) si usa per parole straniere

```
Il calendario più provocatorio è \textit{Sexy Crash},  
il nuovo calendario per camionisti.
```

Il calendario più provocatorio è *Sexy Crash*, il nuovo  
calendario per camionisti.

## Differenza tra `\emph` e `\textit`

È importante separare i due ruoli logici del corsivo e dell'enfaticizzato:

```
\textit{L'uomo primitivo \textit{non} conosceva il bar.}
```

*L'uomo primitivo non conosceva il bar.*

```
\emph{L'uomo primitivo \emph{non} conosceva il bar.}
```

*L'uomo primitivo non conosceva il bar.*

## Uso del grassetto e del sottolineato

Il grassetto (*boldface*) si usa quasi esclusivamente per titoli di paragrafi o sezioni del documento

Per favore, `\textbf{NON}` usatelo nel testo di un documento

Per favore, **NON** usatelo nel testo di un documento

## Uso del maiuscoletto

Il maiuscoletto (*small caps*) si usa solo in bibliografia ed eccezionalmente per i nomi

```
L'insegna \textsc{Bar Sport} era molto bella, e  
il padrone del bar, Antonio detto Onassis, l'aveva  
pagata sessantamila lire nel lontano '65.
```

L'insegna BAR SPORT era molto bella, e il padrone del bar, Antonio detto Onassis, l'aveva pagata sessantamila lire nel lontano '65.

## Uso di *typewriter*

Lo stile "macchina da scrivere" (*typewriter*) si usa per scrivere codice e comandi

Il `\textit{database}` dei pacchetti di `\LaTeX` deve essere rigenerato con il comando `\texttt{texhash}`

Il *database* dei pacchetti di `LATEX` deve essere rigenerato con il comando `texhash`

## Scrivere un indirizzo web

Per gli indirizzi web è conveniente utilizzare il comando `\url`

visitate il nostro sito web all'indirizzo:

```
\url{http://www.guit.sssup.it}
```

visitate il nostro sito web all'indirizzo:

```
http://www.guit.sssup.it
```

Attenzione!

Se si vuole trasformare l'indirizzo in un link, è necessario caricare il pacchetto `hyperref`

## Rimpicciolire il carattere

Per rimpicciolire il carattere si usano i seguenti comandi racchiusi tra due "{" "}" o posizionati all'interno di ambienti

```
\normalsize  
\small  
\footnotesize  
\scriptsize  
\tiny
```

```
tonno  
tonno  
tonno  
tonno  
tonno
```

## Ingrandire il carattere

Per aumentare il carattere si usano i seguenti comandi racchiusi tra due "{" "}" o posizionati all'interno di ambienti

```
\normalsize  
\large  
\Large  
\LARGE  
\huge  
\Huge
```

```
tonno  
tonno  
tonno  
tonno  
tonno  
tonno
```

## Ingrandire il carattere

Se inserito nel testo il comando avrà effetto da quel punto fino alla fine del documento

```
La dialettica escatologica come cura \Large per le emorroidi
```

La dialettica escatologica come cura per le emorroidi

Attenzione!

È buona norma non far variare inutilmente la dimensione del carattere all'interno di blocchi di testo

# Un esempio vale più di mille parole

`esempio_2_8.tex`

## Oggetti flottanti

In  $\text{\LaTeX}$  esiste la possibilità di inserire oggetti (figure o tabelle) esattamente nel punto in cui essi sono posizionati nel testo.

Tuttavia questo è **da evitare**, perché una delle peculiarità di  $\text{\LaTeX}$  è la capacità di inserire oggetti nel posto giudicato ottimale in base a precisi canoni tipografici.

Per questo tabelle e figure sono detti **oggetti mobili o flottanti** (*floating*).

## Oggetti flottanti

Posizionata all'interno di un oggetto flottante l'opzione:

[htb]

Esprime la nostra preferenza circa la posizione nella pagina ove l'oggetto debba essere posizionato.

- **h** posizionale 'qui' (*here*)
- **t** oppure posizionale 'in cima' (*top*)
- **b** o ancora possibile posizionale 'in fondo' (*bottom*);
- **p** o eventualmente posizionale su una pagina dedicata a tutti gli oggetti *float*;

Naturalmente l'ordine è modificabile

## La tabulazione

```
\begin{tabular}{lr}  
  bianco & 102,5 \\  
  nero & 15,4 \\  
\end{tabular}
```

bianco	102,5
nero	15,4

# La tabulazione

```
\begin{tabular}{lr} \hline  
bianco & 102,5 \\  
nero & 15,4 \\ \hline  
\end{tabular}
```

bianco	102,5
nero	15,4

# La tabulazione

```
\begin{tabular}{lr} \hline  
\multicolumn{2}{c}{intestazione} \\ \hline  
bianco & 102,5 \\ nero & 15,4 \\ \hline  
\end{tabular}
```

intestazione	
bianco	102,5
nero	15,4

## Uso del *pipe*

Se proprio non potete evitare di inserire le righe verticali, usate il comando `|` (*pipe*)

```
\begin{tabular}{|l|r|} \hline
\multicolumn{2}{|c|}{intestazione} \\ \hline
bianco & 102,5 \\
nero & 15,4 \\ \hline
\end{tabular}
```

intestazione	
bianco	102,5
nero	15,4

# La tabulazione

```
\begin{center}
\begin{tabular}{lr} \hline
\multicolumn{2}{c}{intestazione} \\ \hline
bianco & 102,5 \\
nero & 15,4 \\ \hline
\end{tabular}
\end{center}
```

<hr/>	
intestazione	
<hr/>	
bianco	102,5
nero	15,4
<hr/>	

## La tabella

```
\begin{table}
\begin{tabular}{lr} \hline
\multicolumn{2}{c}{intestazione} \\ \hline
bianco & 102,5 \\
nero & 15,4 \\ \hline
\end{tabular}
\end{table}
```

<hr/>	
intestazione	
<hr/>	
bianco	102,5
nero	15,4

# La tabella

```
\begin{table}[htb]
\begin{tabular}{lr} \hline
\multicolumn{2}{c}{intestazione} \\ \hline
bianco & 102,5 \\
nero & 15,4 \\ \hline
\end{tabular}
\end{table}
```

<hr/>	
intestazione	
<hr/>	
bianco	102,5
nero	15,4

## La didascalia

La didascalia si inserisce con il comando `\caption`, che come i comandi di sezionamento offre la possibilità di specificare un titolo breve per l'indice delle figure.

```
\caption{Titolo della tabella}
```

Per citare l'oggetto flottante è sufficiente inserire una `\label` dopo la `\caption`

## La didascalia della tabella

```
\begin{table}[htb]\caption{titolo}\label{ciccio}
\begin{tabular}{p{.3\textwidth}r} \hline
\multicolumn{2}{c}{intestazione}\ \ \hline
    bianco & 102,5 \ \
    nero & 15,4 \ \ \hline
\end{tabular}
\end{table}
```

# La didascalia della tabella

Ecco come appare la tabella 1

Tabella 1: titolo

intestazione	
bianco	102,5
nero	15,4

# Un esempio vale più di mille parole

`esempio_3_1.tex`

# L'arte della tipografia matematica

Generalmente la scrittura di formule matematiche costituisce la parte più complessa e delicata della stesura di un documento. Proprio in questo particolare ambito,  $\text{\LaTeX}$  offre una qualità tipografica allo stato dell'arte.

La sintassi per la scrittura di formule matematiche non è assolutamente difficile e richiede appena un minimo di pratica.

## Scrivere le formule nel testo

$\text{\LaTeX}$  applica parecchia cura nella spaziatura nelle formule, anche quando esse sono molto semplici. Ecco un cattivo esempio di come non vanno scritte:

Non è vero che  $7+2=9$  e  $7-3=4$ , sono solo bugie.

Non è vero che  $7+2=9$  e  $7-3=4$ , sono solo bugie.

Il modo corretto di scrivere le formule all'interno del testo è quello di inserirle tra due  $\$ \dots \$$

Non è vero che  $\$7+2=9\$$  e  $\$7-3=4\$$  sono solo bugie.

Non è vero che  $7 + 2 = 9$  e  $7 - 3 = 4$ , sono solo bugie.



## Scrivere le formule nel testo

Se si inserisce la formula nel testo  $\LaTeX$  cerca di schiacciarla per non aumentare l'interlinea.

Dopo lunghi studi, Livsi  $\backslash v\{c\}$  Vr  $\backslash o$  dstadt riuscì a dimostrare che poiché  $\sum_{i=1}^n a_i = 3$  la figura di Carducci è imponentemente stagliata nel panorama poetico del suo tempo.

Dopo lunghi studi, Livsič Vrødstadt riuscì a dimostrare che poiché  $\sum_{i=1}^n a_i = 3$  la figura di Carducci è imponentemente stagliata nel panorama poetico del suo tempo.

## Centrare le formule

Per centrare la formula su una riga occorre inserirla tra un doppio `$$... $$`. In questo caso lo sviluppo verticale sarà maggiore

Livsi `\v{c}` Vr `\o dstadt` riuscì a dimostrare che poiché `$$\sum_{i=1}^n a_i$$` la figura di Carducci è imponentemente stagliata nel panorama poetico del suo tempo.

Livsič Vrødstadt riuscì a dimostrare che poichè

$$\sum_{i=1}^n a_i = 3$$

la figura di Carducci è imponentemente stagliata nel panorama poetico del suo tempo.



## Centrare le formule

Per centrare la formula su una riga si usa l'ambiente `displaymath`

Dimostrare che la formula:

```
\begin{displaymath}
  \prod \theta^{\psi} - 45x
\end{displaymath}
```

non significa assolutamente nulla.

Dimostrare che la formula:

$$\prod \theta^{\psi} - 45x$$

non significa assolutamente nulla.



# Un esempio vale più di mille parole

`esempio_3_4.tex`

# Esponenti

Per inserire un esponente si usa il comando  $\wedge$  (accento circonflesso o *circum*)

```
 $x^y$ 
```

 $x^y$ 

Nel caso di esponenti più complessi si ricorre alle parentesi

```
 $x^{y+1}$ 
```

 $x^{y+1}$

## Indici

Per inserire un indice si usa il comando `_` (*underscore*)

```
$x_y$
```

$$x_y$$

Nel caso di indici multipli si ricorre alle parentesi annidate

```
x_{n_{k_{i}}}
```

$$x_{n_{k_i}}$$

I caratteri diventano via via sempre più piccoli

# Frazioni

Per inserire una frazione si usa il comando `\frac`

```
\begin{displaymath}  
  \frac{1}{a+1}  
\end{displaymath}
```

$$\frac{1}{a+1}$$

# Frazioni

Il comando `\frac` può anche essere annidato

```
\begin{displaymath}
  \frac{x+\frac{1}{x}}{y+\frac{1}{y}}
\end{displaymath}
```

$$\frac{x + \frac{1}{x}}{y + \frac{1}{y}}$$

# Radici

Per scrivere la radice si usa il comando `\sqrt`

```
\begin{displaymath}  
  \sqrt[n+1]{\chi + x}  
\end{displaymath}
```

$$\sqrt[n+1]{\chi + x}$$

# Sommatorie

Il simbolo di sommatoria si scrive con il comando `\sum`

```
\begin{displaymath}
  \sum_{i=1}^{\infty}
\end{displaymath}
```

$$\sum_{i=1}^{\infty}$$

# Limiti

I limiti si scrivono con il comando `\lim`

```
\begin{displaymath}  
  \lim_{i \to \infty}  
\end{displaymath}
```

$$\lim_{i \rightarrow \infty}$$

# Un esempio vale più di mille parole

`esempio_3_5.tex`

# Parentesi automatiche

Per ottenere delle parentesi che si adattano alle dimensioni di quello che contengono si usa `\left(` e `\right)` e analogamente per quadre e graffe.

## Attenzione

Le graffe sono un carattere riservato quindi si scrive `\left\{` e `\right\}`

Da utilizzare per elementi di 'grosse' dimensioni quando non se ne conosce la dimensione (matrici, casi, etc)

# Parentesi

Posso usare parentesi di diverse dimensioni:

```
( x )
\bigl( x \bigr)
\Bigl( x \Bigr)
\biggr( x \biggr)
\Biggr( x \Biggr)
```

$$\begin{array}{c} (x) \\ (x) \\ (x) \\ (x) \\ (x) \end{array}$$

Da utilizzare per le formule ordinarie

## Parentesi

Un esempio di parentesi grande

```
\begin{displaymath}  
  \Biggl(\frac{1}{n+1}\Biggr)^2  
\end{displaymath}
```

$$\left(\frac{1}{n+1}\right)^2$$

Ovviamente `\Bigl` accetta anche parentesi quadre e graffe

# Alcune lettere greche

Scrivere lettere greche all'interno di ambienti matematici è estremamente semplice

```
\alpha
```

```
\beta
```

```
\xi
```

```
\Xi
```

```
\gamma
```

```
\Gamma
```

```
\omega
```

```
\Omega
```

 $\alpha$  $\beta$  $\xi$  $\Xi$  $\gamma$  $\Gamma$  $\omega$  $\Omega$

## Realizzare la bibliografia

La bibliografia è una lista di pubblicazioni generalmente usati per preparare un documento.

$\text{\LaTeX}$  permette di gestire con grandissima efficienza sia la bibliografia che i riferimenti ad essa presenti nel testo, il tutto in accordo con diverse regole bibliografiche adottate.

In  $\text{\LaTeX}$  esistono due metodi per realizzare la bibliografia.

# L'ambiente thebibliography

```
\begin{thebibliography}{666}
  \bibitem{VBF}
  A.~Vitali, L.~Buzzanca, P.~Franco,
  \textit{Fisica dei Quanti},
  un approccio basato sulla teoria di Kirchhoff
\end{thebibliography}
```

## Bibliografia

[1] A. Vitali, L. Buzzanca, P. Franco, *Fisica dei Quanti*, un approccio basato sulla teoria di Kirchhoff

## Citazioni

```
Oggi nel 2010, il nostro paese è invidiato e temuto,  
anche se è tuttora accerchiato dai centri sociali,  
dall'Europa bolscevica e dai molli americani del  
primo presidente ex nero Michael Jackson \cite{VBF}.
```

Oggi nel 2010, il nostro paese è invidiato e temuto, anche se è tuttora accerchiato dai centri sociali, dall'Europa bolscevica e dai molli americani del primo presidente ex nero Michael Jackson [1].

# Un esempio vale più di mille citazioni

`esempio_3_10.tex`

## Figure ed immagini

$\text{\LaTeX}$  prevede sia la possibilità di produrre figure per proprio conto che di inserire figure esterne prodotte da altri programmi. In quest'ultimo caso si segue una metodica diversa dai comuni editor WYSIWYG:

- le figure rimangono in *file separati*, cioè la figura non va “incollata” nel documento ma è sufficiente scrivere un collegamento ad essa.

Risulta quindi conveniente:

- creare nella directory di lavoro una cartella `img`
- salvare in questa directory tutti i file da inserire nel nostro documento finale.

## Vantaggi e svantaggi

### Vantaggi:

- se i contenuti delle figure vengono cambiati è sufficiente sostituire i file e ricompilare. Il nuovo documento generato avrà tutte le figure aggiornate
- la procedura è molto stabile e non crea brutte sorprese

### Svantaggi:

- serve un minimo di esperienza per una piena padronanza del meccanismo
- richiede l'uso di programmi in grado di lavorare in modo sinergico con  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

## Estensioni supportate

Esistono due principali tipi di figure:

Vettoriali (generalmente indicate per i grafici)

- `.pdf` (portable document file)
- `.eps` (encapsulated postscript)
- `.ps` (postscript)

Bitmap (generalmente indicate per immagini)

- `.png` (portable network graphics)
- `.jpeg` o `.jpg` (joint photographic experts group)
- `.tiff` o `.tif` (tagged image file format)

## Estensioni supportate

pdf $\LaTeX$  supporta direttamente file con le seguenti estensioni:

- `.pdf` (portable document file)
- `.png` (portable network graphics)
- `.jpeg` (encapsulated postscript)
- `.tiff` (tagged image file format)

Gli altri formati (`.eps`, `.ps`) andranno convertiti in `.pdf` se non si vorrà usare un differente compilatore.

## L'oggetto del nostro gioco



lion.png

### Attenzione!

È necessario caricare il pacchetto `graphicx` (con l'opzione `pdftex` nel caso in cui si usi pdfL<sup>A</sup>T<sub>E</sub>X come compilatore)



# Il pacchetto graphicx

```
\includegraphics{lion}
```



Il bello di  $\LaTeX$

È consigliabile non specificare l'estensione del file



## Il pacchetto graphicx

È possibile indicare nel preambolo una specifica cartella dove sono contenute tutte le figure utilizzate all'interno del documento

```
\graphicspath{{<path>}}
```

Alternativamente è possibile specificarlo singolarmente all'interno di ogni singolo `\includegraphics`

```
\includegraphics{<path>lion}
```

# Un esempio vale più di mille figure

`esempio_4_1.tex`

# Il pacchetto graphicx

```
\fbox{\includegraphics{lion}}
```



# Il pacchetto graphicx

```
\includegraphics[scale=0.5]{lion}
```



# Il pacchetto graphicx

```
\includegraphics[scale=1.5]{lion}
```



# Il pacchetto graphicx

```
\includegraphics[width=20mm]{lion}
```



## Il pacchetto graphicx

```
\includegraphics[width=20mm, height=40mm]{lion}
```



## Il pacchetto graphicx

```
\includegraphics[width=20mm, height=40mm,  
keepaspectratio]{lion}
```



## Il pacchetto graphicx

```
\includegraphics[angle=-45]{lion}
```



# Il pacchetto graphicx

```
\includegraphics[angle=-45, width=40mm]{lion}
```



## Il pacchetto graphicx

```
\fbox{\includegraphics[angle=-45, width=40mm]{lion}}
```



## Come affrontare (e superare) i problemi

Nei vent'anni di  $\text{\LaTeX}$  sono state sviluppate soluzioni in grado di soddisfare le più impensate esigenze tipografiche. È quindi estremamente improbabile che un problema non sia già stato affrontato e risolto.

In qualunque difficoltà vi troviate sappiate che, a differenza di molti editor WYSIWYG, la soluzione esiste *quasi* sempre ( $\text{\LaTeX}$  non è infatti ancora in grado di preparare un caffè decente).

Occorre solo trovarla. . .

## Prima di tutto

È assolutamente indispensabile leggere una guida di base. Tra i tanti testi liberi in rete, quello consigliato è:



Oetiker, Tobias.

*Una (mica tanto) breve introduzione a  $\text{\LaTeX} 2_{\epsilon}$ .*

<http://www.ctan.org/tex-archive/info/lshort/italian/>

## Identificare il problema

Il  $\text{\LaTeX}$  si presentano due tipi generali di problemi:

- errori di compilazione: si manifestano quando per un errore nel codice il compilatore non riesce a generare l'output
- personalizzare il documento: richiede l'installazione del pacchetto specifico (quale?) o una conoscenza minima del linguaggio a basso livello

## Errori di compilazione

È assolutamente inevitabile commettere errori di scrittura del codice. Per evitarli e correggerli è opportuno:

- formattare in maniera pulita il codice
- leggere sempre il log del compilatore che spesso riporta il numero della riga dell'errore
- correggere un errore appena si presenta

*L<sup>A</sup>T<sub>E</sub>X, purtroppo, non può far nulla per rendere migliori i contenuti.*

Rosa Gini