

L^AT_EX

<https://universibo.ing.unibo.it/latex>
tntimo@libero.it

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is available on the World Wide Web at <http://www.gnu.org/licenses/fdl.html>.

Indice

Introduzione	2		
$\text{T}_{\text{E}}\text{X}$	3		
$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$	4		
Lavorare con $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$	4		
Cosa serve	5		
Come lavorare con $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$	6		
Formati dei file	7		
Caratteristiche dei formati	8		
Alternative	9		
Scrittura del testo	9		
Documento minimo	10		
Documento un po' meno minimo	11		
Parametri della classe	12		
warning bad box	13		
Scrittura del testo	14		
Caratteri	15		
Font	16		
Allineamento	17		
Liste 1	18		
Liste 2	19		
Riferimenti 1	20		
		Riferimenti 2	21
		Lunghezze	22
		Tabelle 1	23
		Tabelle 2	24
		Tabelle 3	25
		Note a piè di pagina	26
		Struttura di un documento	26
		Capitoli, sezioni, ...	27
		Struttura del documento: indici, titoli, ...	28
		Ambienti fluttuanti 1	29
		Ambienti fluttuanti 2	30
		Matematica	30
		Modi matematici 1	31
		Modi matematici 2	32
		Modi matematici 3	33
		Spazi	34
		Esponenti & Pedici	35
		Frazioni	36
		Parentesi 1	37
		Parentesi 2	38
		Simboli	39
		Vettori e altri accenti	40
		Simboli "grandi"	41
		Matrici	42
		Font in modo matematico	43
		Funzioni	44
		Altri ambienti 1	45
		Altri ambienti 2	46
		Altri ambienti di allineamento	47
		L'ambiente array	48
		Immagini	48
		immagini	49
		Formati immagini	50
		Conversioni tra formati	51
		Scrittura del testo – elementi un po' più avanzati	51
		Paragrafi e interlinea	52
		Lingue	53
		Comandi 1	54
		Comandi 2	55
		Ambienti	56
		Dividere il documento in più file	57
		Altri pacchetti	57
		Margini	58
		Numeri	59
		Unità di misura	60

1) $\text{T}_{\text{E}}\text{X}$

- ➊ Nel 1977, Donald Knuth realizza $\text{T}_{\text{E}}\text{X}$, con l'obiettivo di impaginare articoli di matematica evitando che venissero stravolti graficamente in tipografia;

1) **T_EX**

- ❶ Nel 1977, Donald Knuth realizza T_EX, con l'obiettivo di impaginare articoli di matematica evitando che venissero stravolti graficamente in tipografia;
- ❷ sviluppa quindi un **linguaggio** per descrivere l'impaginazione di un documento.

1) **T_EX**

- ❶ Nel 1977, Donald Knuth realizza T_EX, con l'obiettivo di impaginare articoli di matematica evitando che venissero stravolti graficamente in tipografia;
- ❷ sviluppa quindi un linguaggio per descrivere l'impaginazione di un documento.
- ❸ T_EX è un linguaggio di “basso livello”, nel senso che l'utente deve preoccuparsi di tutti gli aspetti dell'impaginazione del documento: grafica, indici,

1) $\text{T}_{\text{E}}\text{X}$

- ❶ Nel 1977, Donald Knuth realizza $\text{T}_{\text{E}}\text{X}$, con l'obiettivo di impaginare articoli di matematica evitando che venissero stravolti graficamente in tipografia;
- ❷ sviluppa quindi un linguaggio per descrivere l'impaginazione di un documento.
- ❸ $\text{T}_{\text{E}}\text{X}$ è un linguaggio di “basso livello”, nel senso che l'utente deve preoccuparsi di tutti gli aspetti dell'impaginazione del documento: grafica, indici,
- ❹ È un linguaggio programmabile: permette la definizione di **macro**.

1) **T_EX**

- ❶ Nel 1977, Donald Knuth realizza T_EX, con l'obiettivo di impaginare articoli di matematica evitando che venissero stravolti graficamente in tipografia;
- ❷ sviluppa quindi un linguaggio per descrivere l'impaginazione di un documento.
- ❸ T_EX è un linguaggio di "basso livello", nel senso che l'utente deve preoccuparsi di tutti gli aspetti dell'impaginazione del documento: grafica, indici,
- ❹ È un linguaggio programmabile: permette la definizione di macro.
- ❺ Il nome deriva dalle prime 3 lettere della parola "τεχνε" (technè) = arte & tecnica; la χ viene sostituita dalla X per indicare che T_EX nasce con un occhio di riguardo per i testi matematici; la pronuncia corretta è **tech**.

2) \LaTeX

- ① Nel 1985, Leslie Lamport sviluppa una raccolta di macro che chiama \LaTeX , attualmente alla versione $\text{\LaTeX} 2_{\epsilon}$:

2) \LaTeX

- ① Nel 1985, Leslie Lamport sviluppa una raccolta di macro che chiama \LaTeX , attualmente alla versione $\text{\LaTeX} 2_{\epsilon}$:
- ② ambiente di “alto livello”, in cui l’utente ha a disposizione comandi per descrivere la struttura **logica** del documento.

2) \LaTeX

- ① Nel 1985, Leslie Lamport sviluppa una raccolta di macro che chiama \LaTeX , attualmente alla versione $\text{\LaTeX} 2_{\epsilon}$:
- ② ambiente di “alto livello”, in cui l’utente ha a disposizione comandi per descrivere la struttura **logica** del documento.
- ③ Sono trasparenti all’utente tutti i dettagli grafici, come il tipo e la dimensione dei font, i margini, la formattazione dei titoli e degli indici, . . .

2) \LaTeX

- ① Nel 1985, Leslie Lamport sviluppa una raccolta di macro che chiama \LaTeX , attualmente alla versione $\text{\LaTeX} 2_{\epsilon}$:
- ② ambiente di “alto livello”, in cui l’utente ha a disposizione comandi per descrivere la struttura **logica** del documento.
- ③ Sono trasparenti all’utente tutti i dettagli grafici, come il tipo e la dimensione dei font, i margini, la formattazione dei titoli e degli indici, . . .
- ④ inoltre sono automatizzate alcune procedure come la generazione degli indici, “riferimenti incrociati”, la numerazione di capitoli, figure, equazioni, . . .

2) \LaTeX

- ① Nel 1985, Leslie Lamport sviluppa una raccolta di macro che chiama \LaTeX , attualmente alla versione $\text{\LaTeX} 2_{\epsilon}$:
- ② ambiente di “alto livello”, in cui l’utente ha a disposizione comandi per descrivere la struttura **logica** del documento.
- ③ Sono trasparenti all’utente tutti i dettagli grafici, come il tipo e la dimensione dei font, i margini, la formattazione dei titoli e degli indici, . . .
- ④ inoltre sono automatizzate alcune procedure come la generazione degli indici, “riferimenti incrociati”, la numerazione di capitoli, figure, equazioni, . . .
- ⑤ I comandi di \LaTeX permettono di indicare dove inizia un capitolo, una sezione, dove finisce un paragrafo, dove inserire un riferimento, un indice, . . .

3) Cosa serve per lavorare con L^AT_EX

- ① Un editor di testo, per scrivere il sorgente;

3) Cosa serve per lavorare con L^AT_EX

- ❶ Un editor di testo, per scrivere il sorgente;
- ❷ il compilatore T_EX;

3) Cosa serve per lavorare con L^AT_EX

- ❶ Un editor di testo, per scrivere il sorgente;
- ❷ il compilatore T_EX;
- ❸ file necessari per la compilazione: font e librerie utilizzate (almeno quelle base di L^AT_EX);

3) Cosa serve per lavorare con L^AT_EX

- ❶ Un editor di testo, per scrivere il sorgente;
- ❷ il compilatore T_EX;
- ❸ file necessari per la compilazione: font e librerie utilizzate (almeno quelle base di L^AT_EX);
- ❹ un **viewer**, in grado di aprire e stampare il file ottenuto.

4) Come lavorare con L^AT_EX

- ① Rispetto agli editor di testo tradizionali, L^AT_EX non è un ambiente WYSIWYG (What You See Is What You Get):

4) Come lavorare con L^AT_EX

- ① Rispetto agli editor di testo tradizionali, L^AT_EX non è un ambiente WYSIWYG (What You See Is What You Get):
- ② non è un grosso svantaggio se si pensa che l'obiettivo è descrivere la **struttura logica** del documento e non la grafica.

4) Come lavorare con L^AT_EX

- ① Rispetto agli editor di testo tradizionali, L^AT_EX non è un ambiente WYSIWYG (What You See Is What You Get):
- ② non è un grosso svantaggio se si pensa che l'obiettivo è descrivere la struttura logica del documento e non la grafica.
- ③ è un vantaggio in termini di flessibilità:
 - a) dallo stesso sorgente si possono ottenere documenti in **formati diversi**
 - b) siccome la resa grafica è basata su stili e classi, cambiandoli si ottengono **rese diverse** del documento
 - c) utilizzando la possibilità di **definire macro**, basta ridefinirle per cambiare l'aspetto del risultato finale.

5) Formati dei file

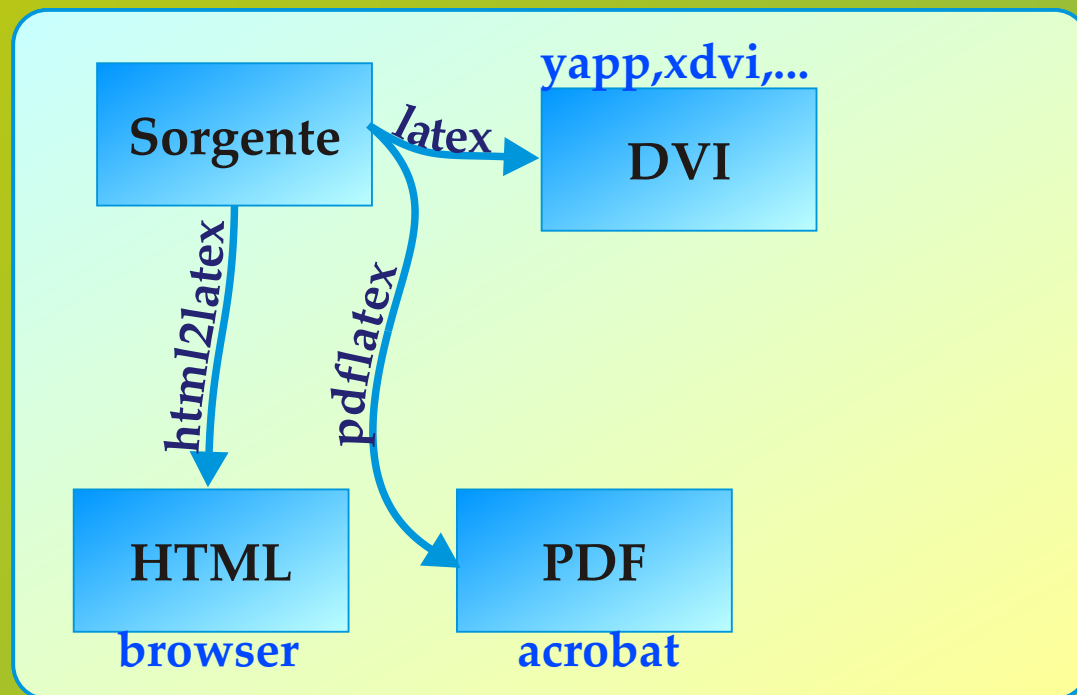
⇒ Il comando `latex` compila il sorgente, producendo un file in formato **DVI** (DeVice Independent).

⇒ Altri comandi permettono di ottenere altri formati (PDF, HTML, RTF, ...).



5) Formati dei file

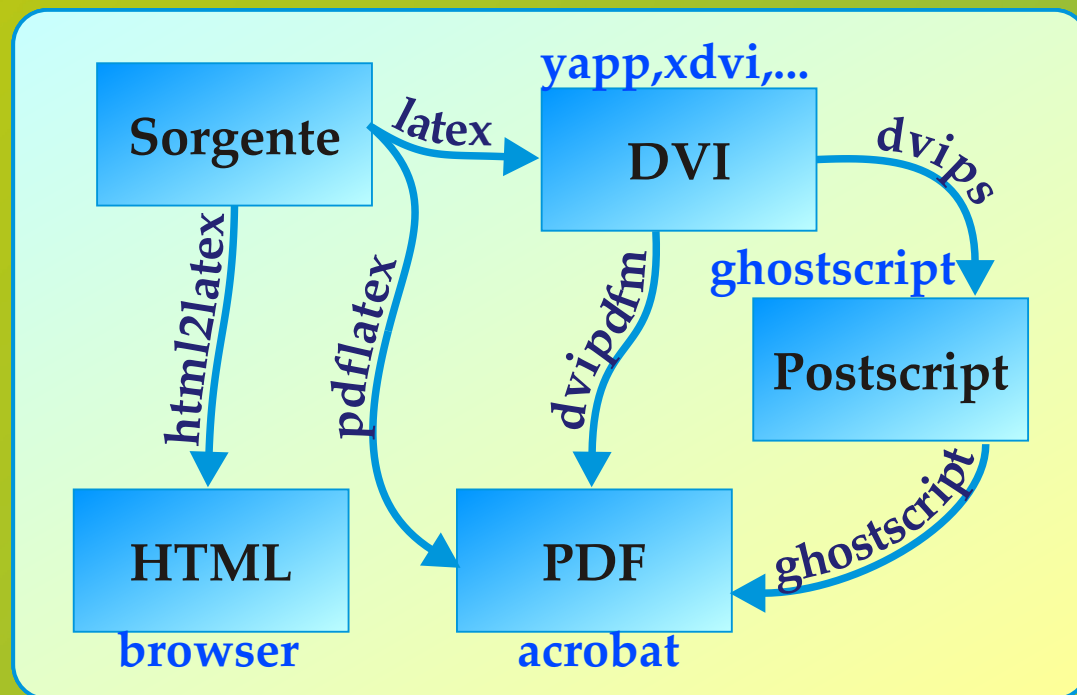
- ⇒ Il comando `latex` compila il sorgente, producendo un file in formato DVI (DeVice Independent).
- ⇒ Altri comandi permettono di ottenere altri formati (PDF, HTML, RTF, ...).



5) Formati dei file

⇒ Il comando `latex` compila il sorgente, producendo un file in formato DVI (DeVice Independent).

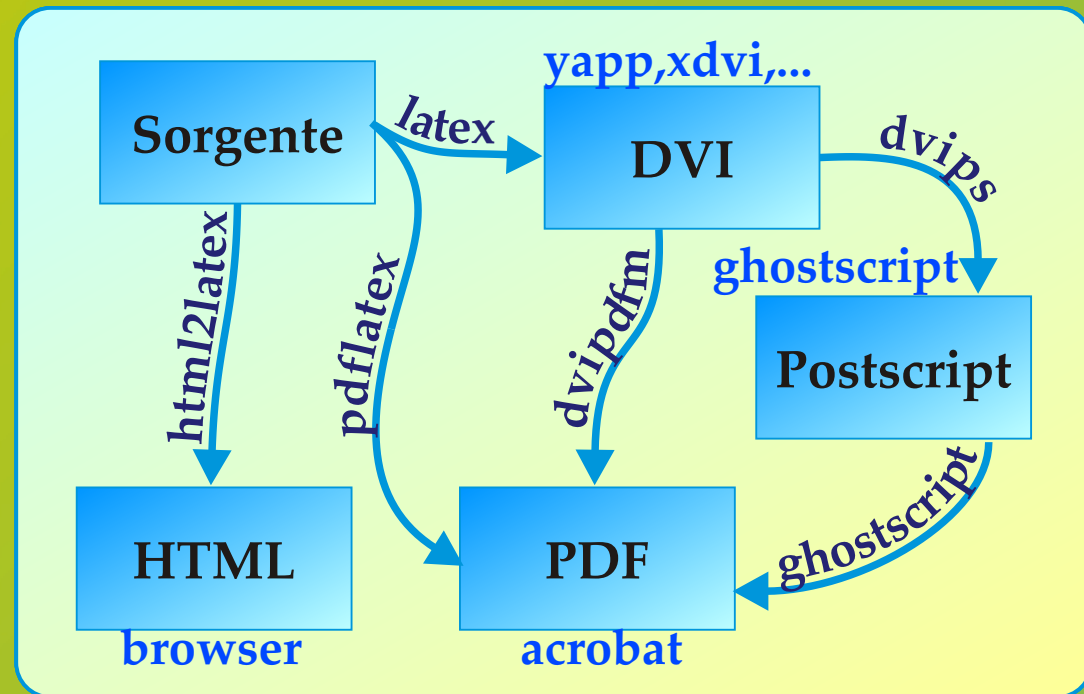
⇒ Altri comandi permettono di ottenere altri formati (PDF, HTML, RTF, ...).



⇒ È possibile poi passare da un formato all'altro con opportuni programmi.

5) Formati dei file

- ⇒ Il comando `latex` compila il sorgente, producendo un file in formato DVI (DeVice Independent).
- ⇒ Altri comandi permettono di ottenere altri formati (PDF, HTML, RTF, ...).



- ⇒ Il formato DVI è molto leggero, perché non include le immagini e i font. Postscript[®] e PDF possono includere tutto ciò che serve in un unico file.
- ⇒ È possibile poi passare da un formato all'altro con opportuni programmi.

6) Caratteristiche dei formati

DVI

- ✓ File piccoli; compilazione molto veloce
- ✓ Possibilità di ricerca da un punto del documento al punto corrispondente del codice e viceversa \Rightarrow molto comodo durante la stesura del documento.
- ✗ Non include file esterni

6) Caratteristiche dei formati

DVI

- ✓ File piccoli; compilazione molto veloce
- ✓ Possibilità di ricerca da un punto del documento al punto corrispondente del codice e viceversa ⇒ molto comodo durante la stesura del documento.
- ✗ Non include file esterni

PDF (prodotto con pdflatex)

- ✗ Compilazione più lenta
- ✓ Tutto incluso in un unico file; comprime testo e immagini ⇒ adatto per distribuire il documento
- ✓ Comodo per la consultazione a schermo: funzioni di ricerca, collegamenti ipertestuali, barra dell'indice, effetti per presentazioni, ...

7) **Alternative**

Esiste uno strumento quasi WYSIWYG, chiamato LyX:

- ① Permette di lavorare in un ambiente grafico, che dà un'idea del risultato che si otterrà (si autodefinisce WYSIWYM – What You See Is What You Mean);
- ② nasconde tutte le caratteristiche di L^AT_EX,
- ③ ma il documento finale viene ottenuto producendo un file L^AT_EX e compilandolo.
- ④ È di fatto alternativo a L^AT_EX, anche se è possibile introdurre direttamente comandi L^AT_EX.
- ⑤ Il documento può essere esportato come sorgente L^AT_EX.

8) Documento minimo

```
\documentclass{report}  
\begin{document}  
  Testo del documento ...  
\end{document}
```

❶ Si dichiara la **classe** con il comando `\documentclass`

Le classi standard sono book, report, article, letter, slide.

8) Documento minimo

```
\documentclass{report}  
\begin{document}  
  Testo del documento ...  
\end{document}
```

❶ Si dichiara la classe con il comando `\documentclass`

Le classi standard sono `book`, `report`, `article`, `letter`, `slide`.

❷ Si scrive il testo all'interno dell'ambiente `document`.

9) Documento un po' meno minimo

```
\documentclass[a4paper,12pt]{report}
\usepackage{amsmath} % Questo è un commento!
\usepackage[T1]{fontenc}
\usepackage[italian]{babel}
\begin{document} ... \end{document}
```

- ① Il comando `\usepackage` carica un **pacchetto**: una libreria contenente la definizione di macro;

9) Documento un po' meno minimo

```
\documentclass[a4paper,12pt]{report}
\usepackage{amsmath} % Questo è un commento!
\usepackage[T1]{fontenc}
\usepackage[italian]{babel}
\begin{document} ... \end{document}
```

- ① Il comando `\usepackage` carica un pacchetto: una libreria contenente la definizione di macro;
- ② `amsmath`: funzioni avanzate per la scrittura di formule.
- ③ `babel`: adatta \LaTeX a lingue diverse dall'inglese.
- ④ `fontenc`: permette l'utilizzo di caratteri come à è ì ò ù.

9) Documento un po' meno minimo

```
\documentclass[a4paper,12pt]{report}
\usepackage{amsmath} % Questo è un commento!
\usepackage[T1]{fontenc}
\usepackage[italian]{babel}
\begin{document} ... \end{document}
```

- ① Il comando `\usepackage` carica un pacchetto: una libreria contenente la definizione di macro;
- ② `amsmath`: funzioni avanzate per la scrittura di formule.
- ③ `babel`: adatta \LaTeX a lingue diverse dall'inglese.
- ④ `fontenc`: permette l'utilizzo di caratteri come à è ì ò ù.
- ⑤ Il carattere `%` inizia un **commento**.

9) Documento un po' meno minimo

```
\documentclass[a4paper,12pt]{report}
\usepackage{amsmath} % Questo è un commento!
\usepackage[T1]{fontenc}
\usepackage[italian]{babel}
\begin{document} ... \end{document}
```

- ① Il comando `\usepackage` carica un pacchetto: una libreria contenente la definizione di macro;
- ② `amsmath`: funzioni avanzate per la scrittura di formule.
- ③ `babel`: adatta \LaTeX a lingue diverse dall'inglese.
- ④ `fontenc`: permette l'utilizzo di caratteri come à è ì ò ù.
- ⑤ Il carattere `%` inizia un commento.
- ⑥ La parte di sorgente prima di `\begin{document}` è detta **preambolo**.

10) Parametri della classe

⑦ Alcuni dei parametri che le classi standard accettano:

10pt

dimensione base del font – sono ammessi i valori 10, 11, 12

a4paper

dimensione della pagina

twoside

impaginazione per stampa fronte-retro

fleqn

formule allineate a sinistra invece che al centro

leqno

numeri delle formule a sinistra invece che a destra

10) Parametri della classe

⑦ Alcuni dei parametri che le classi standard accettano:

10pt

dimensione base del font – sono ammessi i valori 10, 11, 12

a4paper

dimensione della pagina

twoside

impaginazione per stampa fronte-retro

fleqn

formule allineate a sinistra invece che al centro

leqno

numeri delle formule a sinistra invece che a destra

⑧ È possibile specificare parametri globali che verranno usati da altri pacchetti, ad esempio

italian

draft che evidenzia i warning del tipo bad hbox con un rettangolo nero – inoltre fa sì che i comandi del pacchetto **graphicx** inseriscano dei rettangoli vuoti al posto delle immagini.

11) `warning bad box`

⇒ “`Overfull \hbox (56.81487pt too wide) in paragraph at lines 62--70`”: indica una riga troppo lunga nel paragrafo che si trova alle linee 62–70 del sorgente `LATEX`; la riga sborda di $56,814 \cdot 87$ punti ($\approx 2\text{cm}$).

11) **warning bad box**

- ⇒ “Overfull `\hbox` (56.81487pt too wide) in paragraph at lines 62--70”: indica una riga troppo lunga nel paragrafo che si trova alle linee 62–70 del sorgente \LaTeX ; la riga sborda di $56,814 \cdot 87$ punti ($\approx 2\text{cm}$).
- ⇒ “Underfull `\hbox` (badness 1067) in paragraph at lines 10--10”: indica una riga che è stata terminata senza che il motore di allineamento riuscisse a riempirla, nel paragrafo che si trova alla linea 10 del sorgente \LaTeX , a cui è stato assegnato un *voto* di 1067 in una scala da 0 a $10 \cdot 000$, dove 0 è una riga perfetta e $10 \cdot 000$ il punteggio peggiore.

12) Scrittura del testo

- ✓ I caratteri di spaziatura (spazi, tabulazioni, interruzioni di linea) per L^AT_EX sono equivalenti,
- ✓ spazi multipli vengono sempre ridotti ad uno spazio solo.

12) Scrittura del testo

- ✓ I caratteri di spaziatura (spazi, tabulazioni, interruzioni di linea) per L^AT_EX sono equivalenti,
- ✓ spazi multipli vengono sempre ridotti ad uno spazio solo.
- ✓ Il testo è suddiviso in **paragrafi**: la fine di un paragrafo viene indicata dal comando. `\par` o da una linea lasciata vuota.

12) Scrittura del testo

- ✓ I caratteri di spaziatura (spazi, tabulazioni, interruzioni di linea) per L^AT_EX sono equivalenti,
- ✓ spazi multipli vengono sempre ridotti ad uno spazio solo.
- ✓ Il testo è suddiviso in paragrafi: la fine di un paragrafo viene indicata dal comando `\par` o da una linea lasciata vuota.
- ✓ Si può inserire una **interruzione di linea** senza interrompere un paragrafo con `\\` oppure `\newline`.

```
Paragrafo      1.\par Un altro
paragrafo che dovrebbe andare a
capo.
```

```
Parag\\rafo 3.
```

```
Paragrafo 1.
```

```
Un altro paragrafo
che dovrebbe andare a
capo.
```

```
Parag
rafo 3.
```

13) Caratteri

❶ Alcuni caratteri che fanno parte della sintassi si ottengono

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

13) Caratteri

❶ Alcuni caratteri che fanno parte della sintassi si ottengono

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

❷ Fanno eccezione: `\` che si ottiene ad esempio con `\backslash`, `~` che si può ottenere con `\sim`.

13) Caratteri

- ❶ Alcuni caratteri che fanno parte della sintassi si ottengono
`\$ \& \% \# _ \{ \}`
- ❷ Fanno eccezione: `\` che si ottiene ad esempio con `$$backslash$`, `~` che si può ottenere con `$$sim$`.
- ❸ Le “virgolette” si fanno “virgolette”, dove il primo è il carattere ASCII 96 e il secondo è l’apostrofo.

13) **Caratteri**

❶ Alcuni caratteri che fanno parte della sintassi si ottengono

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

❷ Fanno eccezione: `\` che si ottiene ad esempio con `$$backslash$`, `~` che si può ottenere con `$$sim$`.

❸ Le “virgolette” si fanno “virgolette”, dove il primo è il carattere ASCII 96 e il secondo è l’apostrofo.

❹ I caratteri **accentati** si ottengono con particolari combinazioni:

`\'o` ó `\'o` ò `\~o` ã `\=o` õ `\.o` ò `\"o` ö `\r` Å Å

Per la ì si deve scrivere `\'\i` altrimenti si otterrebbe ì

13) Caratteri

❶ Alcuni caratteri che fanno parte della sintassi si ottengono

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

❷ Fanno eccezione: `\` che si ottiene ad esempio con `$$backslash$`, `~` che si può ottenere con `$$sim$`.

❸ Le “virgolette” si fanno “virgolette”, dove il primo è il carattere ASCII 96 e il secondo è l’apostrofo.

❹ I caratteri accentati si ottengono con particolari combinazioni:

`\'o ó \'o ò \~o õ \=o ô \.o ò \"o ö \r A Å`

Per la `ì` si deve scrivere `\'i` altrimenti si otterrebbe `ï`

➤ i comandi seguenti sono **fontenc** Usando **fontenc** si possono usare i caratteri ASCII estesi (quindi le lettere accentate si inseriscono semplicemente digitandole così come sono sulla tastiera).

14) **Font**

❶ Comandi per lo stile del testo:

`\emph{...}` *enfaticizzato*`\textbf{...}` **grassetto**`\texttt{...}` type writer`\textsf{...}` sans serif`\textsc{...}` SMALL CAPS`\textit{...}` *italic*

14) Font

❶ Comandi per lo stile del testo:

<code>\emph{...}</code>	<i>enfaticizzato</i>	<code>\textbf{...}</code>	grassetto
<code>\texttt{...}</code>	type writer	<code>\textsf{...}</code>	sans serif
<code>\textsc{...}</code>	SMALL CAPS	<code>\textit{...}</code>	<i>italic</i>

Ricordando che la filosofia di \LaTeX è rappresentare la logica del documento senza preoccuparsi della sua resa grafica, sarebbe bene usare il più possibile solo il primo di questi comandi.

14) **Font**

❶ Comandi per lo stile del testo:

<code>\emph{...}</code>	<i>enfaticizzato</i>	<code>\textbf{...}</code>	grassetto
<code>\texttt{...}</code>	type writer	<code>\textsf{...}</code>	sans serif
<code>\textsc{...}</code>	SMALL CAPS	<code>\textit{...}</code>	<i>italic</i>

Ricordando che la filosofia di \LaTeX è rappresentare la logica del documento senza preoccuparsi della sua resa grafica, sarebbe bene usare il più possibile solo il primo di questi comandi.

❷ Dimensione del testo:

<code>\Huge</code>	<code>\huge</code>	<code>\LARGE</code>	<code>\Large</code>
<code>\large</code>	<code>\normalsize</code>	<code>\footnotesize</code>	<code>\tiny</code>

Tutti questi comandi sono attivi dal punto in cui sono scritti fino alla fine dell'ambiente o delle parentesi `{ }` in cui si trovano:

<pre>testo {\large grande} --- normale</pre>	testo grande — normale
--	-------------------------------

15) **Allineamento**

- ① Solitamente il testo viene **giustificato** – cioè tutte le righe hanno la stessa larghezza;

15) Allineamento

- ① Solitamente il testo viene giustificato – cioè tutte le righe hanno la stessa larghezza;
- ② per farlo gli spazi tra le parole vengono opportunamente allargati; \LaTeX utilizza una variabile per indicare la tolleranza di questo adattamento. Con il comando `\sloppy` si aumenta la tolleranza, mentre con `\fussy` si abbassa (quest'ultima è l'impostazione di default).

15) Allineamento

- ① Solitamente il testo viene giustificato – cioè tutte le righe hanno la stessa larghezza;
- ② per farlo gli spazi tra le parole vengono opportunamente allargati; \LaTeX utilizza una variabile per indicare la tolleranza di questo adattamento. Con il comando `\sloppy` si aumenta la tolleranza, mentre con `\fussy` si abbassa (quest'ultima è l'impostazione di default).
- ③ Paragrafi con allineamenti diversi:

```
\begin{flushleft} sinistra  
\end{flushleft}  
\begin{center} centro  
\end{center}  
\begin{flushright} destra  
\end{flushright}
```

sinistra

centro

destra

16) **Liste 1**

① **Liste non numerate**: si ottengono con l'ambiente `itemize`;

Testo:

```
\begin{itemize}
  \item elemento
  \item lista annidata:
    \begin{itemize}
      \item primo
      \item secondo
    \end{itemize}
  \item[ $\clubsuit$ ] punto diverso
\end{itemize}
```

Testo:

- elemento
- lista annidata:
 - * primo
 - * secondo
- ♣ punto diverso

17) Liste 2

② Liste numerate: si ottengono con l'ambiente `enumerate`.

17) **Liste 2**

② Liste numerate: si ottengono con l'ambiente `enumerate`.

③ Liste con descrizione: ambiente `description`.

Esempio di lista numerata:

```
\begin{enumerate}
```

```
  \item primo
```

```
  \item secondo
```

```
\end{enumerate}
```

Altro esempio; liste:

```
\begin{description}
```

```
  \item[numerate] ambiente
  enumerate
```

```
  \item[con descrizione]
```

```
ambiente description
```

```
\end{description}
```

Esempio di lista numerata:

1. primo

2. secondo

Altro esempio; liste:

numerate ambiente `enume-
rate`

con descrizione ambiente
`description`

18) Riferimenti 1

① Con il comando `\label{nome}` si inserisce un'etichetta;

18) Riferimenti 1

- ① Con il comando `\label{nome}` si inserisce un'etichetta;
- ② con i comandi `\ref{nome}`, `\pageref{nome}` si inseriscono dei **riferimenti** all'etichetta.

18) Riferimenti 1

- ① Con il comando `\label{nome}` si inserisce un'etichetta;
- ② con i comandi `\ref{nome}`, `\pageref{nome}` si inseriscono dei riferimenti all'etichetta.
- ③ Il compilatore alla prima passata scrive in un file le posizioni delle etichette, alla seconda utilizza questi dati per inserire i riferimenti, per cui per essere certi che i riferimenti siano corretti bisogna **lanciarlo 2 volte di seguito**. Se i riferimenti non sono corretti, viene prodotto un messaggio di warning che lo segnala – ad esempio “Label(s) may have changed”.

18) Riferimenti 1

- ① Con il comando `\label{nome}` si inserisce un'etichetta;
- ② con i comandi `\ref{nome}`, `\pageref{nome}` si inseriscono dei riferimenti all'etichetta.
- ③ Il compilatore alla prima passata scrive in un file le posizioni delle etichette, alla seconda utilizza questi dati per inserire i riferimenti, per cui per essere certi che i riferimenti siano corretti bisogna lanciarlo 2 volte di seguito. Se i riferimenti non sono corretti, viene prodotto un messaggio di warning che lo segnala – ad esempio “Label(s) may have changed”.
- ④ Come i comandi, anche i nomi delle etichette sono case-sensitive; se un'etichetta non viene trovata il compilatore produce il warning: “There were undefined references”.

19) Riferimenti 2

- ⑤ ➤ i comandi seguenti sono `varioref` Fornisce il comando `\vref{nome}`, che inserisce il riferimento all'etichetta aggiungendo anche la pagina, se l'etichetta si trova in una pagina diversa da quella in cui compare il riferimento.

19) Riferimenti 2

- ⑤ ➤ i comandi seguenti sono `varioref` Fornisce il comando `\vref{nome}`, che inserisce il riferimento all'etichetta aggiungendo anche la pagina, se l'etichetta si trova in una pagina diversa da quella in cui compare il riferimento.

```
A pag. \pageref{storiaLaTeX}
abbiamo accennato alla storia di
\LaTeX{}.
```

```
\begin{equation}
  \label{eq:banalità}
  1+1=2
\end{equation}
```

Vedi le equazioni
`\vref{eq:banalità}`,
`\vref{eq:esempioEq}` nel lucido
`\ref{slide:modiMat2}`.

A pag. 4 abbiamo accennato alla storia di \LaTeX .

$$1 + 1 = 2 \quad (1)$$

Vedi le equazioni 1, 2 a pagina 32 nel lucido 30.

20) Lunghezze


Alcuni comandi richiedono come parametri delle **lunghezze**; possono essere inserite come:

- 1 un **numero** seguito da una **unità** di misura, es. `3.6cm`; le unità riconosciute sono:

in pollice	pt punto, $1\text{pt} = \frac{1}{72,27}\text{in}$
cm centimetro	mm millimetro
ex altezza di un x	em larghezza di una M

- 2 memorizzate in una variabile, utilizzando i comandi `\newlength`, `\setlength`, `\addtolength`, `\settowidth`, `\settoheight`.

Variabili di uso frequente sono `\textwidth`, `\textheight`.

- 3  i comandi seguenti sono forniti dal pacchetto `calc` è possibile fare operazioni tra lunghezze, come `.8\textwidth-3ex`.

21) **Tabelle 1**

① Le **tabelle** vengono costruite dall'ambiente **tabular**,

21) Tabelle 1

- ① Le tabelle vengono costruite dall'ambiente `tabular`,
- ② che richiede un parametro con l'indicazione delle **colonne**: ognuna è rappresentata da una lettera che indica l'allineamento: `l`, `c`, `r`.

21) Tabelle 1

- ① Le tabelle vengono costruite dall'ambiente `tabular`,
- ② che richiede un parametro con l'indicazione delle colonne: ognuna è rappresentata da una lettera che indica l'allineamento: `l`, `c`, `r`.
- ③ Le **celle** di una riga sono separate dal carattere `&`;

21) Tabelle 1

- ① Le tabelle vengono costruite dall'ambiente `tabular`,
- ② che richiede un parametro con l'indicazione delle colonne: ognuna è rappresentata da una lettera che indica l'allineamento: `l`, `c`, `r`.
- ③ Le celle di una riga sono separate dal carattere `&`;
- ④ La fine di una `linea` della tabella è indicato da `\\`.

21) **Tabelle 1**

- ① Le tabelle vengono costruite dall'ambiente `tabular`,
- ② che richiede un parametro con l'indicazione delle colonne: ognuna è rappresentata da una lettera che indica l'allineamento: `l`, `c`, `r`.
- ③ Le celle di una riga sono separate dal carattere `&`;
- ④ La fine di una linea della tabella è indicato da `\\`.

```
\dots tabella:
\begin{center}
\begin{tabular}{crl}
$1^a$ & colonna a &
questa è a \\
colonna & destra &
sinistra
\end{tabular}
\end{center}
```

```
... tabella:
      1a      colonna a   questa è a
colonna      destra sinistra
```

22) Tabelle 2

- ⑤ Nella definizione delle colonne si possono indicare le **righe verticali** con il carattere **||**

22) Tabelle 2

- ⑤ Nella definizione delle colonne si possono indicare le righe verticali con il carattere `||`
- ⑥ Le righe orizzontali sono inserite dal comando `\hline`

22) **Tabelle 2**

- ⑤ Nella definizione delle colonne si possono indicare le righe verticali con il carattere `|`
- ⑥ Le righe orizzontali sono inserite dal comando `\hline`
- ⑦ Le colonne del tipo `l`, `c`, `r`, formattano sempre il testo su un'unica linea anche se è molto lungo (e non è possibile inserire dei `\\`); esiste anche la colonna `p{larghezza}`, che ha *larghezza fissata e allineamento giustificato*.

```
\begin{tabular}{|c|p{10ex}|}
\hline
colonna&colonna\\
\hline
su una sola linea&
che invece va anche a
capo\\
\hline
\end{tabular}
```

colonna	colonna
su una sola linea	che invece va anche a capo

23) **Tabelle 3**

⑧ Si possono creare **celle su più colonne** con il comando

```
\multicolumn{colonne}{allineamento}{testo}
```

```
\begin{tabular}{|1|1|} \hline  
\multicolumn{2}{|c|}{2  
colonne}\\\hline  
colonna 1&colonna2\\\hline  
\end{tabular}
```

2 colonne	
colonna 1	colonna2

23) **Tabelle 3**

⑧ Si possono creare celle su più colonne con il comando `\multicolumn{colonne}{allineamento}{testo}`

```
\begin{tabular}{|1|1|} \hline
\multicolumn{2}{|c|}{2
colonne}\\\hline
colonna 1&colonna2\\\hline
\end{tabular}
```

2 colonne	
colonna 1	colonna2

► i comandi seguenti sono **longtable** Fornisce un ambiente **longtable**, identico a **tabular**, ma in grado di spezzare automaticamente le tabelle lunghe su più pagine. Se viene specificato un titolo con il comando `\caption{titolo}` viene inserito anche nell'indice delle tabelle (vedi slide 27). Inoltre tramite i comandi (da usare al posto `\\`) `\endhead`, `\endfirsthead`, `\endfoot`, `\endlastfoot`, si possono specificare una o più righe da inserire all'inizio della tabella, all'inizio di ogni pagina (tranne la 1^a), alla fine di ogni pagina (tranne l'ultima), alla fine della tabella.

24) Note a piè di pagina

- ✓ Le note a piè di pagina si inseriscono con il comando `\footnote{testo della nota}`.

Questo è un primo\footnote{Nota del primo paragrafo.}
paragrafo.

Anche il secondo\footnote{Quante note in questa pagina!}
paragrafo ha una nota.

Questo è un primo¹ paragrafo.

Anche il secondo² paragrafo ha una nota.

¹ Nota del primo paragrafo.

² Quante note in questa pagina!

25) Capitoli, sezioni, ...

- ❶ Un documento è suddiviso in varie parti, con 6 livelli di suddivisione:
 1. part
 2. chapter
 3. section
 4. subsection
 5. paragraph
 6. subparagraph
- ❷ I capitoli sono definiti solo nelle classi `book`, `report`.

25) Capitoli, sezioni, ...

- ❶ Un documento è suddiviso in varie parti, con 6 livelli di suddivisione:
 1. part
 2. chapter
 3. section
 4. subsection
 5. paragraph
 6. subparagraph
- ❷ I capitoli sono definiti solo nelle classi `book`, `report`.
- ❸ Tutti questi comandi hanno sintassi:
`\parte[voce indice]{titolo}`; il parametro facoltativo, se specificato, è il testo che comparirà nell'indice al posto del titolo.

25) Capitoli, sezioni, ...

- ❶ Un documento è suddiviso in varie parti, con 6 livelli di suddivisione:
 1. part
 2. chapter
 3. section
 4. subsection
 5. paragraph
 6. subparagraph
- ❷ I capitoli sono definiti solo nelle classi `book`, `report`.
- ❸ Tutti questi comandi hanno sintassi:
`\parte[voce indice]{titolo}`; il parametro facoltativo, se specificato, è il testo che comparirà nell'indice al posto del titolo.
- ❹ Esiste la versione *asteriscata* `\parte*{titolo}` che produce una parte non numerata e che non figura nell'indice.

25) Capitoli, sezioni, ...

- ❶ Un documento è suddiviso in varie parti, con 6 livelli di suddivisione:
 1. part
 2. chapter
 3. section
 4. subsection
 5. paragraph
 6. subparagraph
- ❷ I capitoli sono definiti solo nelle classi `book`, `report`.
- ❸ Tutti questi comandi hanno sintassi:
`\parte[voce indice]{titolo}`; il parametro facoltativo, se specificato, è il testo che comparirà nell'indice al posto del titolo.
- ❹ Esiste la versione *asteriscata* `\parte*{titolo}` che produce una parte non numerata e che non figura nell'indice.
- ❺ Per creare una appendice `\appendix\chapter{titolo}`.

26) Struttura del documento: indici, titoli, ...

```
\begin{document}
  \author{autore}
  \title{titolo}
  \date{data} % se non specificata viene messa quella odierna
  \maketitle % genera il frontespizio - pagina del titolo
  \tableofcontents % genera l'indice
  \chapter{Primo Capitolo}
  ...
\end{document}
```

27) Ambienti fluttuanti 1

- ❶ Oggetti “grandi” come tabelle e figure, non possono essere inseriti in mezzo al testo, ma vengono posizionati dove c’è posto. \LaTeX fornisce due ambienti per inserire elementi floating: `figure`, `table`

27) Ambienti fluttuanti 1

❶ Oggetti “grandi” come tabelle e figure, non possono essere inseriti in mezzo al testo, ma vengono posizionati dove c’è posto. \LaTeX fornisce due ambienti per inserire elementi floating: `figure`, `table`

❷ È possibile specificare alcune preferenze sulla posizione dell’oggetto:

```
\begin{figure}[!htbp]
  \centering
  ...
  \caption[voce
indice]{titolo}
  \label{fig:etichetta}
\end{figure}
```

h qui, se c’è posto
t in cima a una pagina
b in fondo a una pagina
p in una pagina di soli float
! forza il compilatore

28) Ambienti fluttuanti 2

- ③ L'indice delle figure viene generato con il comando `\listoffigures`; quello delle tabelle con `\listoftables`

28) Ambienti fluttuanti 2

- ③ L'indice delle figure viene generato con il comando `\listoffigures`; quello delle tabelle con `\listoftables`
- ④ Il comando `\clearpage` inserisce una interruzione di pagina, forzando il compilatore ad inserire tutti i float non ancora inseriti nel documento; utile per evitare che alcuni float vengano inseriti troppe pagine oltre il punto del testo a cui si riferiscono.

28) Ambienti fluttuanti 2

- ③ L'indice delle figure viene generato con il comando `\listoffigures`; quello delle tabelle con `\listoftables`
- ④ Il comando `\clearpage` inserisce una interruzione di pagina, forzando il compilatore ad inserire tutti i float non ancora inseriti nel documento; utile per evitare che alcuni float vengano inseriti troppe pagine oltre il punto del testo a cui si riferiscono.
- ⑤ ➤ i comandi seguenti sono `afterpage`
L'uso migliore è `\afterpage{\clearpage}`, che esegue il comando appena finisce la pagina corrente, evitando l'interruzione di una pagina non ancora completata.

29) Modi matematici 1

- ① Per scrivere formule matematiche bisogna passare dal modo testo al **modo matematico**: la prima possibilità è usare l'ambiente `displaymath`, che dispone una formula su una nuova linea centrandola nella pagina; una abbreviazione per questo ambiente è `\[\]`;

29) Modi matematici 1

- ❶ Per scrivere formule matematiche bisogna passare dal modo testo al modo matematico: la prima possibilità è usare l'ambiente `displaymath`, che dispone una formula su una nuova linea centrandola nella pagina; una abbreviazione per questo ambiente è `\[\]`;
- ❷ per introdurre formule "in linea" nel testo si racchiude la formula tra `$ $`.

Le formule in linea come
`$E_c=\frac{1}{2}mv^2$` sono più
 piccole
`\[`
`E_c=\frac{1}{2}mv^2`
`\]`
 di quelle scritte così.

Le formule in linea come
 $E_c = \frac{1}{2}mv^2$ sono più piccole

$$E_c = \frac{1}{2}mv^2$$

di quelle scritte così.

30) Modi matematici 2

- ③ L'ambiente `equation` inserisce **formule numerate**; identico a `displaymath` – solitamente contiene una `\label`:

```
\begin{equation}
  \label{eq:esempioEq}
  f(x)=\int_0^x
    \sqrt{\left|\sin\left(
      \frac{1}{\xi}\right)\right|}d\xi \end{equation}
```

$$f(x) = \int_0^x \sqrt{\left|\sin\left(\frac{1}{\xi}\right)\right|} d\xi \quad (2)$$

31) Modi matematici 3

④ Il comando `\text{...}` inserisce ... in modo testo:

```
\[f(x)\triangleq\left\{
  \begin{aligned}
    2x & \quad x\geq 0\\
    0 & \quad \text{altrimenti}
  \end{aligned}
\right.\]
```

$$f(x) \triangleq \begin{cases} 2x & x \geq 0 \\ 0 & \text{altrimenti} \end{cases}$$

32) Spazi

① Gli spazi in modo matematico vengono ignorati.

32) Spazi

① Gli spazi in modo matematico vengono ignorati.

② È possibile forzare l'inserimento di uno spazio usando alcune combinazioni:

`\!` □□ spazio negativo

`\,` □□

`\;` □□

`\quad` □ □

`\qquad` □ □

33) Esponenti & Pedici

① Gli esponenti si ottengono con il carattere \wedge ;

33) Esponenti & Pedici

- ① Gli esponenti si ottengono con il carattere `^`;
- ② i **pedici** con il carattere `_`.

33) Esponenti & Pedici

- ① Gli esponenti si ottengono con il carattere `^`;
- ② i pedici con il carattere `_`.
- ③ Se l'apice (o il pedice) è un più di un carattere o un simbolo, deve racchiuso tra `{ }`.
- ④ Se si devono mettere sia un apice che un pedice, vanno indicati in sequenza.

```
\[x^2_i - x^\star\  
\[e^{\-\frac{3}{2} x}\]
```

$$x_i^2 - x^\star$$

$$e^{-\frac{3}{2}x}$$

34) **Frazioni**

① Si usa il comando `\frac{numeratore}{denominatore}`.

34) Frazioni

- ① Si usa il comando `\frac{numeratore}{denominatore}`.
- ② \blacktriangleright \mathcal{AMS} La frazione viene inserita grande o piccola a discrezione di \LaTeX ; si può forzare l'inserimento di una frazione come apparirebbe in un ambiente `displaymath` usando `\dfrac`, o come apparirebbe in un ambiente `$$` usando `\tfrac`.

Nel testo: `$$\frac{1}{2}$$` oppure `$$\dfrac{1}{2}$$`; mentre in `displaymath`:

```
\[
  \frac{1}{2}e^{\frac{x-y}{2}}
  \neq \tfrac{1}{2}
  e^{\dfrac{x-y}{2}}
\]
```

Nel testo: $\frac{1}{2}$ oppure $\frac{1}{2}$; mentre in `displaymath`:

$$\frac{1}{2}e^{\frac{x-y}{2}} \neq \frac{1}{2}e^{\frac{x-y}{2}}$$

35) Parentesi 1

① Le parentesi utilizzabili sono

()	(...)		<code>\langle</code>	<code>\rangle</code>	<code>\langle</code>	<code>\rangle</code>
[]	[...]	► <i>AMS</i>	<code>\lvert</code>	<code>\rvert</code>	<code> </code>	<code>\cdots</code>
{ }	{...}	► <i>AMS</i>	<code>\lVert</code>	<code>\rVert</code>	<code>\ </code>	<code>\cdots</code>

35) Parentesi 1

❶ Le parentesi utilizzabili sono

<code>()</code>	<code>(\dots)</code>		<code>\langle</code>	<code>\rangle</code>	<code>\langle \dots \rangle</code>
<code>[]</code>	<code>[\dots]</code>	► <i>AMS</i>	<code>\lvert</code>	<code>\rvert</code>	<code> \dots </code>
<code>{ }</code>	<code>\{\dots\}</code>	► <i>AMS</i>	<code>\lVert</code>	<code>\rVert</code>	<code>\ \dots\ </code>

❷ Per indicare che si devono **adattare alle dimensioni** del contenuto, bisogna scrivere `\left` davanti alle parentesi aperte e `\right` davanti alle parentesi chiuse:

<pre>\[\{\frac{1}{2}\}, \quad \quad \left\{ \frac{1}{2}\right\} \]</pre>	$\frac{1}{2}, \quad \left\{\frac{1}{2}\right\}$
--	---

36) Parentesi 2

③ Inoltre è possibile forzare le dimensione con i prefissi `\Bigg`, `\bigg`, `\Big`, `\big`.

```
\left((x+1)\cdot(y-2)\right)^2
```

$$((x + 1) \cdot (y - 2))^2$$

```
\big((x+1)\cdot(y-2)\big)^2
```

$$\left((x + 1) \cdot (y - 2)\right)^2$$

36) Parentesi 2

- ③ Inoltre è possibile forzare le dimensione con i prefissi `\Bigg`, `\bigg`, `\Big`, `\big`.

```
\left((x+1)\cdot(y-2)\right)^2\]
\big((x+1)\cdot(y-2)\big)^2\]
```

$$((x + 1) \cdot (y - 2))^2$$

$$\left((x + 1) \cdot (y - 2)\right)^2$$

- ④ Ogni comando `\left` deve avere il corrispondente `\right`; si può inserire un delimitatore senza parentesi usando `\left.` oppure `\right.`.

```
\[
\left\{\begin{aligned}
&x+y+z=2 \\
&-x+2y-z=1 \\
&y-z=0
\end{aligned}\right.
\rightarrow \dots
```

$$\begin{cases} x + y + z = 2 \\ y - z = 0 \\ -x + 2y - z = 1 \end{cases} \Rightarrow \begin{cases} x = 0 \\ y = 1 \\ z = 1 \end{cases}$$

37) Simboli

① Alcuni delle centinaia di **simboli** definiti:

\neq	<code>\neq</code>	\leq	<code>\le</code>	\geq	<code>\ge</code>	\equiv	<code>\equiv</code>	\ll	<code>\ll</code>
\gg	<code>\gg</code>	\doteq	<code>\doteq</code>	\sim	<code>\sim</code>	\nsim	<code>\nsim</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\in	<code>\in</code>	\notin	<code>\notin</code>	\propto	<code>\propto</code>	\perp	<code>\perp</code>
∞	<code>\infty</code>	\forall	<code>\forall</code>	\exists	<code>\exists</code>	\hbar	<code>\hbar</code>	\imath	<code>\imath</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	∂	<code>\partial</code>	∇	<code>\nabla</code>	\times	<code>\times</code>
\rightarrow	<code>\rightarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>				

37) Simboli

① Alcuni delle centinaia di simboli definiti:

\neq	<code>\neq</code>	\leq	<code>\le</code>	\geq	<code>\ge</code>	\equiv	<code>\equiv</code>	\ll	<code>\ll</code>
\gg	<code>\gg</code>	\doteq	<code>\doteq</code>	\sim	<code>\sim</code>	\nsim	<code>\nsim</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\in	<code>\in</code>	\notin	<code>\notin</code>	\propto	<code>\propto</code>	\perp	<code>\perp</code>
∞	<code>\infty</code>	\forall	<code>\forall</code>	\exists	<code>\exists</code>	\hbar	<code>\hbar</code>	\imath	<code>\imath</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	∂	<code>\partial</code>	∇	<code>\nabla</code>	\times	<code>\times</code>
<hr/>									
\rightarrow	<code>\rightarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>				

② Lettere greche

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

38) Vettori e altri accenti

① Simboli sopra un singolo carattere:

`\vec{v}` \vec{v} `\bar{v}` \bar{v} `\tilde{v}` \tilde{v} `\dot{v}` \dot{v}

► *AMS* `\overset{a}{b}` $\overset{a}{b}$ `\underset{a}{b}` $\underset{a}{b}$

38) Vettori e altri accenti

❶ Simboli sopra un singolo carattere:

`\vec{v}` \vec{v} `\bar{v}` \bar{v} `\tilde{v}` \tilde{v} `\dot{v}` \dot{v}

➤ *AMS* `\overset{a}{b}` $\overset{a}{b}$ `\underset{a}{b}` $\underset{a}{b}$

❷ Simboli più grandi:

`\overrightarrow{abc}` \overrightarrow{abc} `\widetilde{abc}` \widetilde{abc}

`\widehat{abc}` \widehat{abc} `\overline{abc}` \overline{abc}

38) Vettori e altri accenti

❶ Simboli sopra un singolo carattere:

`\vec{v}` \vec{v} `\bar{v}` \bar{v} `\tilde{v}` \tilde{v} `\dot{v}` \dot{v}

➤ \mathcal{AMS} `\overset{a}{b}` $\overset{a}{b}$ `\underset{a}{b}` $\underset{a}{b}$

❷ Simboli più grandi:

`\overrightarrow{abc}` \overrightarrow{abc} `\widetilde{abc}` \widetilde{abc}

`\widehat{abc}` \widehat{abc} `\overline{abc}` \overline{abc}

❸ ➤ \mathcal{AMS} `\xrightarrow[sotto]{sopra}` $\xrightarrow[sotto]{sopra}$

38) Vettori e altri accenti

❶ Simboli sopra un singolo carattere:

`\vec{v}` \vec{v} `\bar{v}` \bar{v} `\tilde{v}` \tilde{v} `\dot{v}` \dot{v}

➤ \mathcal{AMS} `\overset{a}{b}` $\overset{a}{b}$ `\underset{a}{b}` $\underset{a}{b}$

❷ Simboli più grandi:

`\overrightarrow{abc}` \overrightarrow{abc} `\widetilde{abc}` \widetilde{abc}

`\widehat{abc}` \widehat{abc} `\overline{abc}` \overline{abc}

❸ ➤ \mathcal{AMS} `\xrightarrow[sotto]{sopra}` $\xrightarrow[sotto]{sopra}$

❹ Graffe

orizzontali:

`\underbrace{testo}_{sotto}`

`\overbrace{testo}^{sopra}`

testo	⏟
sotto	
sopra	
testo	⏟

39) Simboli "grandi"

⇒ Simboli che appaiono diversamente in $\$ \$$ e $\llbracket \llbracket$:

\int \oint \sum \prod \bigcup \bigcap

```
\sum_{i=0}^{\infty} P(i), \;
\lim_{x \rightarrow +\infty} f(x), \;
\oint \vec{F} \cdot d\vec{\ell}, \;
►AMS \iiint_A g(x,y,z)
```

Nel testo $\sum_{i=0}^{\infty} P(i)$, $\lim_{x \rightarrow +\infty} f(x)$, $\oint \vec{F} \cdot d\vec{\ell}$, $\iiint_A g(x,y,z)$; mentre in displaymath:

$$\sum_{i=0}^{\infty} P(i), \quad \lim_{x \rightarrow +\infty} f(x), \quad \oint \vec{F} \cdot d\vec{\ell}, \quad \iiint_A g(x,y,z)$$

39) Simboli "grandi"

⇒ Simboli che appaiono diversamente in `$ $` e `\[\]`:

`\int` \int `\oint` \oint `\sum` \sum `\prod` \prod `\bigcup` \cup `\bigcap` \cap

```

\sum_{i=0}^{\infty} P(i), \;
\lim_{x \rightarrow +\infty} f(x), \;
\oint \vec{F} \cdot d\vec{\ell}, \;
►AMS \iiint_A g(x,y,z)
Nel testo \sum_{i=0}^{\infty} P(i), \lim_{x \rightarrow +\infty} f(x), \oint \vec{F} \cdot d\vec{\ell}, \iiint_A g(x,y,z); mentre in
displaymath:

```

$$\sum_{i=0}^{\infty} P(i), \lim_{x \rightarrow +\infty} f(x), \oint \vec{F} \cdot d\vec{\ell}, \iiint_A g(x,y,z)$$

⇒ Estrazione di radice: `\sqrt[y]{x}` $\sqrt[y]{x}$

dal manuale di `amsmath`: `\sqrt[\leftroot{-1}\uproot{2}y]{x}` $\sqrt[y]{x}$

40) **Matrici** \blacktriangleright *AMS*

❶ L'ambiente `matrix` ha un uso simile a `tabular`

<pre>\[\begin{matrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \ddots \\ \end{matrix}\]</pre>	$\begin{matrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \ddots \end{matrix}$
--	--

40) Matrici \rightarrow AMS

❶ L'ambiente `matrix` ha un uso simile a `tabular`

<pre> \[\begin{matrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \ddots \\ \end{matrix}\] </pre>	$ \begin{matrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \ddots \end{matrix} $
--	--

❷ Esistono versioni di questo ambiente con delimitatori:

<code>pmatrix</code>	<code>bmatrix</code>	<code>Bmatrix</code>	<code>vmatrix</code>	<code>Vmatrix</code>
$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\left\{ \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \right\}$	$\left \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \right $	$\left\ \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \right\ $

40) Matrici \blacktriangleright *AMS*

❶ L'ambiente `matrix` ha un uso simile a `tabular`

<pre> \[\begin{matrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \ddots \\ \end{matrix}\] </pre>	$ \begin{matrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \ddots \end{matrix} $
--	--

❷ Esistono versioni di questo ambiente con delimitatori:

<code>pmatrix</code>	<code>bmatrix</code>	<code>Bmatrix</code>	<code>vmatrix</code>	<code>Vmatrix</code>
$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{Bmatrix} 1 & 0 \\ 0 & 1 \end{Bmatrix}$	$\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$	$\begin{Vmatrix} 1 & 0 \\ 0 & 1 \end{Vmatrix}$

❸ Per inserire matrici piccole $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ esiste l'ambiente `smallmatrix`.

41) Font in modo matematico

- ① Le dimensioni del font possono essere impostate con i comandi `\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle`

41) Font in modo matematico

① Le dimensioni del font possono essere impostate con i comandi `\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle`

② In modo matematico è possibile usare vari font:

`\mathcal{...}` \mathcal{ABC} `\mathbb{...}` \mathbb{ABC}
`\mathrm{...}` ABC `\mathfrak{...}` \mathfrak{ABC}

41) Font in modo matematico

- ① Le dimensioni del font possono essere impostate con i comandi `\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle`
- ② In modo matematico è possibile usare vari font:
`\mathcal{...}` \mathcal{ABC} `\mathbb{...}` \mathbb{ABC}
`\mathrm{...}` ABC `\mathfrak{...}` \mathfrak{ABC}
- ③ Si può ottenere il grassetto con `\mathbf{...}`; però funziona solo con certi font (non con il greco ad esempio).
 ► \mathcal{AMS} Se non funziona si può usare `\boldsymbol{...}`:

```
\[\mathbf{A}\quad\boldsymbol{\xi}\]
```

 $A \quad \xi$

42) **Funzioni**

- ✓ Il testo in modo matematico risulta in *corsivo*; gli operatori matematici andrebbero scritti in *roman* (font “diritto”), per distinguerli da variabili e altre funzioni. Sono definiti una serie di comandi:

`\sin` sin `\log` log `\lim` lim ...

42) **Funzioni**

- ✓ Il testo in modo matematico risulta in *corsivo*; gli operatori matematici andrebbero scritti in *roman* (font “diritto”), per distinguerli da variabili e altre funzioni. Sono definiti una serie di comandi:

`\sin` *sin* `\log` *log* `\lim` *lim* ...

- ✓ ➤ \mathcal{AMS} È possibile definire nuovi operatori con (deve essere usato nel preambolo) `\DeclareMathOperator{\nome}{testo}`;

42) Funzioni

- ✓ Il testo in modo matematico risulta in *corsivo*; gli operatori matematici andrebbero scritti in *roman* (font “diritto”), per distinguerli da variabili e altre funzioni. Sono definiti una serie di comandi:

`\sin` sin `\log` log `\lim` lim ...

- ✓ ➤ \mathcal{AMS} È possibile definire nuovi operatori con (deve essere usato nel preambolo) `\DeclareMathOperator{\nome}{testo}`;
- ✓ oppure la versione `\DeclareMathOperator*` produce operatori in cui apici e pedici vengono posizionati sopra e sotto.

```
\DeclareMathOperator{\sen}{sen}
\DeclareMathOperator*{\massimo}{max}
\[\sen^2\theta\quad\massimo_{0\leq y\leq 1}f(y)\]
```

$$\text{sen}^2 \theta \quad \max_{0 \leq y \leq 1} f(y)$$

43) Altri ambienti 1 ► \mathcal{AMS}

- ❶ Ambiente `gather`: **più equazioni**, allineate al centro; ogni equazione viene numerata. Si può eliminare la numerazione scrivendo `\notag` prima della formula.

```
\begin{gather}\label{eq:gather1}
  f(x)=\log_3(x)\\\ \notag x+y-z=5\\
  \label{eq:gather2}1=\sin^2(\theta)+\cos^2(\theta)
\end{gather}
```

$$f(x) = \log_3(x) \quad (3)$$

$$x + y - z = 5$$

$$1 = \sin^2(\theta) + \cos^2(\theta) \quad (4)$$

43) Altri ambienti 1 ➤ \mathcal{AMS}

❶ Ambiente `gather`: più equazioni, allineate al centro; ogni equazione viene numerata. Si può eliminare la numerazione scrivendo `\notag` prima della formula.

```
\begin{gather}\label{eq:gather1}
  f(x)=\log_3(x)\\\ \notag x+y-z=5\\
  \label{eq:gather2}1=\sin^2(\theta)+\cos^2(\theta)
\end{gather}
```

$$f(x) = \log_3(x) \tag{3}$$

$$x + y - z = 5$$

$$1 = \sin^2(\theta) + \cos^2(\theta) \tag{4}$$

❷ La versione `gather*` non numerata le equazioni.

44) Altri ambienti 2 ➔ \mathcal{AMS}

③ Ambiente `align`: identico a `gather`, ma le formule sono **allineate** rispetto ad un carattere `&` (come in una tabella di 2 colonne, la prima allineata a destra e la seconda a sinistra).

```
\begin{align}
  \label{eq:align1} f(x) &= \log_3(x) \\
  \notag x+y-z &= 5 \\
  \label{eq:align2} 1 &= \sin^2(\theta) + \cos^2(\theta)
\end{align}
```

$$f(x) = \log_3(x) \tag{5}$$

$$x + y - z = 5$$

$$1 = \sin^2(\theta) + \cos^2(\theta) \tag{6}$$

44) Altri ambienti 2 \blacktriangleright \mathcal{AMS}

- ③ Ambiente `align`: identico a `gather`, ma le formule sono allineate rispetto ad un carattere `&` (come in una tabella di 2 colonne, la prima allineata a destra e la seconda a sinistra).

```
\begin{align}
  \label{eq:align1} f(x) &= \log_3(x) \\
  \notag x+y-z &= 5 \\
  \label{eq:align2} 1 &= \sin^2(\theta) + \cos^2(\theta)
\end{align}
```

$$f(x) = \log_3(x) \quad (5)$$

$$x + y - z = 5$$

$$1 = \sin^2(\theta) + \cos^2(\theta) \quad (6)$$

- ④ La versione `align*` non numera le equazioni.

45) Altri ambienti di allineamento $\blacktriangleright \mathcal{AMS}$

- ① All'interno di altri ambienti matematici si possono usare gli ambienti `gathered`, `aligned`, già visti in altri esempi.

45) Altri ambienti di allineamento ► \mathcal{AMS}

- ① All'interno di altri ambienti matematici si possono usare gli ambienti `gathered`, `aligned`, già visti in altri esempi.
- ② In più esiste anche l'ambiente `split`, per una singola lunga formula su **più linee**:

```
\begin{equation}\label{eq:split}\begin{split}
e^x|_{x\rightarrow 0}=&1+x+\tfrac{1}{2}x^2+\\
&+\tfrac{1}{6}x^3+\tfrac{1}{24}x^4+\cdots+\\
&+\tfrac{1}{n!}x^n+o(x^n)
\end{split}\end{equation}
```

$$\begin{aligned}
e^x|_{x\rightarrow 0} &= 1 + x + \frac{1}{2}x^2 + \\
&+ \frac{1}{6}x^3 + \frac{1}{24}x^4 + \cdots + \\
&+ \frac{1}{n!}x^n + o(x^n)
\end{aligned} \tag{7}$$

46) L'ambiente array

✓ L'ambiente `array` è un ambiente di basso livello, identico all'ambiente `tabular`, ma da usare in modo matematico.

46) L'ambiente array

✓ L'ambiente `array` è un ambiente di basso livello, identico all'ambiente `tabular`, ma da usare in modo matematico.

```
\[\widetilde{M}=\left[
\begin{array}{ccc|c}
x_{11}&x_{12}&x_{13}&\mathbf{0}^T\\
x_{12}&x_{22}&x_{23}&\mathbf{0}^T\\
x_{13}&x_{23}&x_{33}&\mathbf{0}^T\\
\hline
\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{I}
\end{array}\right]
```

$$\widetilde{M} = \left[\begin{array}{ccc|c} x_{11} & x_{12} & x_{13} & \mathbf{0}^T \\ x_{12} & x_{22} & x_{23} & \mathbf{0}^T \\ x_{13} & x_{23} & x_{33} & \mathbf{0}^T \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array} \right]$$

47) immagini

➤ i comandi seguenti sono `graphicx` (o dal più vecchio `graphics`)
forniti dal pacchetto

① `\includegraphics[parametri]{file}`

47) immagini

➤ i comandi seguenti sono `graphicx` (o dal più vecchio `graphics`)
forniti dal pacchetto

❶ `\includegraphics[parametri]{file}`

❷ Il file dovrebbe essere specificato senza estensione – il driver “sceglie” l’estensione che preferisce.

47) **immagini**

➤ i comandi seguenti sono **graphicx** (o dal più vecchio **graphics**)
forniti dal pacchetto

❶ `\includegraphics[parametri]{file}`

❷ Il file dovrebbe essere specificato senza estensione – il driver “sceglie” l’estensione che preferisce.

```
\includegraphics{cestino}
\includegraphics[scale=1.5,angle=45]{cestino}
\includegraphics[width=3ex,height=4em]{cestino}
\includegraphics[width=5em,angle=90]{cestino}
```



48) **Formati immagini**

pdflatex riconosce i formati: pdf, jpeg, png, tiff.

48) Formati immagini

pdflatex riconosce i formati: pdf, jpeg, png, tiff.

latex non gestisce le immagini; si limita a lasciare il posto per inserirle. È il programma che dovrà elaborare il dvi che le deve manipolare, da questo dipende quali formati è possibile usare. Supponiamo di avere a disposizione dvips.

48) Formati immagini

pdflatex riconosce i formati: pdf, jpeg, png, tiff.

latex non gestisce le immagini; si limita a lasciare il posto per inserirle. È il programma che dovrà elaborare il dvi che le deve manipolare, da questo dipende quali formati è possibile usare. Supponiamo di avere a disposizione dvips.

- il formato preferenziale è eps (Encapsulated PostScript)
- sono utilizzabili alcuni formati bitmap:
 1. per poter includere file ad esempio png, si scrive nel preambolo `\DeclareGraphicsRule{.png}{bmp}{.bb}{}{}`
 2. occorre un file con lo stesso nome ed estensione .bb contenente le dimensioni del file; può essere creato con il programma `ebb`
 3. il viewer (o il programma di conversione se dal dvi vogliamo ottenere un altro formato) utilizzato deve supportare il formato del file (e deve trovare il file)

49) Conversioni tra formati

		formato che si vuole ottenere				
		<i>ps</i>	<i>eps</i>	<i>pdf</i>	<i>jpeg</i>	<i>wmf</i>
<i>ps</i>	stampa su file	"ritagliarlo" con ghostscript	ghostscript	ghostscript	—	
<i>eps</i>	sono la stessa cosa	esportando in questo formato	ghostscript; epstopdf; eps2pdf	ghostscript	—	
<i>pdf</i>	stampa su file	—	pochi programmi possono salvare in pdf	ghostscript	—	
<i>jpeg</i>	stampa su file	jpeg2eps	—	tutti i programmi di grafica lo supportano	—	
<i>wmf</i>	stampa su file	wmf2eps	—	—	formato vettoriale del mondo windows	

50) Paragrafi e interlinea

- ❶ L'interlinea può essere modificata con il comando `\linespread{fattore}`; l'interlinea doppia si ottiene col valore 1,6.

50) Paragrafi e interlinea

- ❶ L'interlinea può essere modificata con il comando `\linespread{fattore}`; l'interlinea doppia si ottiene col valore 1,6.
- ❷ Si può inserire uno spazio tra due paragrafi con:
`\smallskip` `\medskip` `\bigskip`

50) Paragrafi e interlinea

- ❶ L'interlinea può essere modificata con il comando `\linespread{fattore}`; l'interlinea doppia si ottiene col valore 1,6.
- ❷ Si può inserire uno spazio tra due paragrafi con:
`\smallskip \medskip \bigskip`
- ❸ Se si vogliono cambiare tutte le spaziature tra i paragrafi, è possibile assegnare un nuovo valore alla variabile che controlla la distanza tra paragrafi:
`\setlength{\parskip}{lunghezza}`
- ❹ Analogamente si può cambiare il valore dell'indentazione della prima riga di un paragrafo:
`\setlength{\parindent}{lunghezza}`

51) Lingue

➤ i comandi seguenti sono `babel` forniti dal pacchetto

- ① Quando si carica il pacchetto `babel` è possibile specificare più lingue: es. `\usepackage[italian,english]{babel}`
- ② Si può passare da una lingua all'altra con `\selectlanguage{lingua}`
- ③ Per piccole porzioni di testo si può usare `\foreignlanguage{lingua}{testo}`

52) Comandi 1

⇒ I comandi iniziano con il carattere `\`;

⇒ i nomi sono **case-sensitive** e possono contenere solo caratteri alfabetici;

52) Comandi 1

- ⇒ I comandi iniziano con il carattere `\`;
- ⇒ i nomi sono case-sensitive e possono contenere solo caratteri alfabetici;
- ⇒ si possono definire **nuovi comandi** con
`\newcommand{\nome}[n. parametri]{...}`

52) **Comandi 1**

⇒ I comandi iniziano con il carattere `\`;

⇒ i nomi sono case-sensitive e possono contenere solo caratteri alfabetici;

⇒ si possono definire nuovi comandi con

```
\newcommand{\nome}[n. parametri]{...}
```

⇒ Sono ammessi fino a 9 **parametri**, indicati con `#1` ... `#9`, se per il primo è specificato un valore di default, diventa opzionale e si indica tra `[]`

```
\newcommand{\gradiC}{\ensuremath{^\circ C}}
\newcommand{\parteRe}[1]{\{\mathbb{R}\}\mathrm{e}\left\{\#1\right\}}
\newcommand{\email}[2][email.it]{\ttfamily\#2@\#1}
32\gradiC --- $\parteRe{1-2\jmath}$ --- \email{tizio},
\email[dominio.it]{email}
32°C —  $\mathbb{R}\{1-2j\}$  — tizio@email.it, email@dominio.it
```


54) Ambienti

✓ È possibile definire **ambienti** usando

```
\newenvironment{nome}[n. parametri]
```

```
[default]{codice iniziale}{codice finale}
```

✓ Come per i comandi si possono ridefinire gli ambienti con

```
\renewenvironment
```

54) Ambienti

✓ È possibile definire ambienti usando

```
\newenvironment{nome} [n. parametri]
[default]{codice iniziale}{codice finale}
```

✓ Come per i comandi si possono ridefinire gli ambienti con

```
\renewenvironment
```

```
\newenvironment{eq}[1]%
{\begin{equation}
  \label{eq:#1}}{\end{equation}}
Nell'espressione~\ref{eq:esempio}
definiamo $\hbar$:
\begin{eq}{esempio}
  \hbar\dot{=} \frac{h}{2\pi}
\end{eq}
```

Nell'espressione 8 definiamo \hbar :

$$\hbar \doteq \frac{h}{2\pi} \quad (8)$$

55) Dividere il documento in più file

- ✓ Il comando `\input{file}`, permette di includere un file T_EX;
- ✓ è possibile (e consigliabile) spezzare un documento lungo in più file.

```
\documentclass{report}
\begin{document}
  \tableofcontents
  \input{cap1.tex}\input{cap2.tex}
\begin{document}
... cap1.tex .....
\chapter{introduzione}
Qui si scrive il codice, senza inserirlo nell'ambiente
document e senza dichiarare la classe ...
```

56) Margini

① ➤ i comandi seguenti sono forniti dal pacchetto **a4wide** Imposta dei margini più ridotti (quindi la pagina diventa più ampia) adatti al formato a4 – basta caricare il pacchetto.

② ➤ i comandi seguenti sono forniti dal pacchetto **geometry** Caricandolo si possono specificare vari parametri:

```
\usepackage[margin=lunghezza]{geometry}  
\usepackage[text={larghezza,altezza},  
centering]{geometry}
```

57) Numeri

- ❶ ➤ i comandi seguenti sono `numprint` forniti dal pacchetto

```
\numprint{1234.4567e-89} 1 · 234,456 · 7 · 10-89
```

- ❷ purtroppo va adattato manualmente alla propria lingua:

```
\decimalsign{,}\productsign{\cdot}
\thousandsep{\ensuremath{\,,\dot{ }}}}
```

- ❸ fornisce un nuovo tipo di colonna negli ambienti tabular:

```
\begin{tabular}{|n{7}{6}|}\hline
  \numprint{123457.8e2}\\\hline
  \numprint{12.345678e-123}\\\hline
\end{tabular}
```

1 · 234 · 567,8	· 10 ²
12,345 · 678 · 10 ⁻¹²³	

58) Unità di misura

➤ i comandi seguenti sono **SIunits** scrive le unità di misura secondo le specifiche del SI:

```
$1\mu m/s^2$
```

```
$1\micro\metre\per\square\second$
```

```
1$\mu$m/s$^2$
```

```
1\micro\metre\per\square\second
```

```
\numprint{2.2}\angstrom
```

```
20\celsius
```

 $1\mu\text{m}/\text{s}^2$ $1\mu\text{m}/\text{s}^2$
 $1\mu\text{m}/\text{s}^2$ $1\mu\text{m}/\text{s}^2$
 $2,2\text{\AA}$ 20°C