

TeX – Regole di uso comune

Versione preliminare — Commenti e suggerimenti sono molto graditi, e possono essere riportati a M.R. Modenato o G. Benettin.

1. Scrittura di un testo ordinario

Non richiede attenzioni speciali; si scrive tutto di seguito, senza spezzare la parola a fine riga, e TeX impagina. Si possono usare tutti i caratteri della tastiera alfanumerica dell'IBM ed i caratteri ASCII dal 32 al 126, tranne i seguenti che sono

Caratteri di controllo:

- \$: segna l'inizio e fine del linguaggio matematico;
- \ : introduce i comandi. Esempio: \it comanda di passare all'*italic*, \rm comanda di tornare al roman (v. oltre);
- { } : individuano blocchi di testo. Esempio: \rm Nel mezzo { \it del cammin } di nostra vita "*del cammin*" appare in italic; tutto il resto in roman.
- # : usato nelle definizioni di nuovi comandi, v. oltre;
- & : usato per allineamenti (formule, tabelle, etc.);
- ^ _ : usati rispettivamente per soprascritti e sottoscritti nel linguaggio matematico;
- ~ : è equivalente ad uno spazio bianco, ma vieta di andare a caporiga in quel punto. Esempio: D.~Alighieri compare come D. Alighieri, ma TeX non spezzerà mai l'iniziale dal nome.
- % : usato per inserire commenti. TeX ignora tutto quello che segue fino alla fine della riga.

Questi simboli si possono ottenere facendoli precedere da \ (es. \% genera %), ad esclusione del \ stesso che si ottiene scrivendo \backslash (\backslash, \{ e \} funzionano però solo in "linguaggio matematico", v. oltre).

Inoltre:

- < > : funzionano solo in linguaggio matematico;
- § : si ottiene scrivendo \S;
- : è usato nel linguaggio matematico come segno meno; nel testo ordinario è meglio usare il doppio segno meno(--). Esempio: A.A. 1986-87 si avrà scrivendo A.A. 1986--87.

Cambio di paragrafo: una o più righe bianche, oppure \par. Il paragrafo nuovo comincia rientrato.

Spaziatura verticale: \vskip (misura): es. \vskip 1.2 cm salta 1.2 cm. Le unità di misura disponibili sono cm, mm, in(=pollici), pt(=punti tipografici; le maiuscole ordinarie sono alte 10 punti), ed anche altre.

Spaziatura orizzontale: spazi di misura crescente sono \ (spazio), \quad, \quadquad.

Fonti di caratteri: sono disponibili \rm (roman), \it (*italic*), \bf (**boldface, grassetto**), \sl (*slanted, inclinato, ma meno riccioluto dell'italic*). Per altre fonti di carattere (in particolare, fonti ingrandite) si rimanda al manuale.

Accenti: grave ed acuto: \ ' e \ ', prima della vocale da accentare: perch\ 'e, ci\ 'o, \ 'e. Con la i è conveniente sopprimere il puntino; la i senza puntino si scrive \i, e si accenta cos\ 'i. Attenzione: per ragioni di carattere generale, \i si mangia gli spazi bianchi successivi, che (se necessario) vanno imposti con \ (spazio). Per altri tipi di accenti si rimanda al manuale.

Virgolette: si aprono con “(doppio carattere 96 ASCII), si chiudono con ”(doppio apostrofo da tastiera), oppure con ” (virgolette da tastiera).

Sottolineature: si ottengono con il comando \underbar; ad esempio, importante si scrive \underbar{importante}.

Controllo della pagina:

\hsize=12 cm: fissa la dimensione orizzontale a 12 cm*

\vsize=15.5 cm: fissa la dimensione verticale a 15.5 cm*

\hoffset=± 12 mm: sposta il testo a destra (+) o sinistra (-) di 12 mm

\voffset=± 2 cm: sposta il testo in basso (+) o alto (-) di 2 cm

\baselineskip=30 pt: fissa l'interlinea a 30 pt (interlinea doppia)

\parskip=2 mm: fissa lo spazio tra paragrafi a 2 mm (ordinariamente vale zero)

\parindent=1 cm: fissa di 1 cm il rientro all'inizio dei paragrafi

\noindent: sopprime il rientro per il paragrafo immediatamente successivo

\nopagenumbers: sopprime la numerazione delle pagine

\centerline { Bibliografia }: scrive Bibliografia in centro

\vfill, seguito da \eject: riempie di righe bianche la pagina e fa iniziare il testo seguente all'inizio della pagina successiva

* il vanilla style prevede solo: \pagewidth {dimensioni}e \pageheight {dimensioni}, rispettivamente.

Ingrandimenti: si mette, in testa, \magnification=\magstep1,, =\magstep5. Ogni step aumenta le dimensioni lineari del 20%. Attenzione: tutte le misure in cm, mm, etc. risultano anch'esse ingrandite. Esistono però unità “vere”, che non ingrandiscono: truecm, truemm, truept, etc. Perciò \hsize=12 truecm resta 12 cm con qualunque \magnification.

Fine lavoro: alla fine del testo, per chiudere si deve scrivere \bye.

2. Linguaggio matematico

Inizia e termina con `$` per le formule in riga; inizia e termina con `$$` per le formule in evidenza. Esempio: `$ x = 1$` genera $x = 1$ nel testo; `$$ x = 1 $$` lo mette in evidenza. In questo caso si suggerisce di scrivere, per chiarezza,

`$$`

$x = 1$

`$$`

e il risultato è:

$$x = 1$$

Attenzione: $x = 1$ è diverso da $\$ x = 1 \$$. Nel linguaggio matematico valgono regole speciali, ad esempio:

- ★ tutte le lettere sono in italic
- ★ i numeri sono in roman
- ★ la spaziatura segue criteri diversi dal testo ed interamente automatici
- ★ alcuni comandi funzionano solo in modo matematico (es. `\sum`, v. oltre)

Testi del tipo: ove x, y, z e t sono inferiori a a , si scrivono ove $\$x\$, \$y\$, \$z\$$ e $\$t\$$ sono inferiori a $\$a\$$. Non scrivere: $\$x, y, z\$$ e $\$t\$$; viene male, ed inoltre `TEX` va in difficoltà a spezzare le righe.

EspONENTI ed INDICI: $\{...\}$ e $\{...\}$ portano all'esponente ed a piede il contenuto della parentesi. X_{ij}^{3n+1} si scrive `X_{ij}^{3n+1}` (l'ordine di apice e pedice è irrilevante). Altro esempio: $X_1^{n_1+m^2}$ si scrive `X_{1}^{n_{1}+m^{2}}`. Quando entro parentesi vi sia un solo simbolo, la parentesi si può omettere: la scrittura precedente è equivalente a $X_{1}^{n_{1}+m^{2}}$.

Invece $X^{(2n+1)}y$ è un errore, perchè genera $X^{(2n+1)}y$ (solo la parentesi è stata portata ad esponente).

Frazioni: a piacere, `\frac {numeratore }{denominatore}`, oppure `{numeratore \over denominatore}`. Esempio:

`\frac {X^{2n}}{y^{2n}} = (\frac{x}{y})^{2n}` genera $\frac{X^{2n}}{y^{2n}} = (\frac{x}{y})^{2n}$. La stessa cosa si ottiene scrivendo così: `{x^{2n}\over y^{2n}} = ({x\over y})^{2n}`. Nel primo modo (ma non nel secondo, pena catastrofi) si possono omettere le `{}` che contengono un solo carattere: `$ \frac 12 $` genera $\frac{1}{2}$.

Simboli vari: ogni simbolo ha un nome. Es.: `\alpha, \beta, \gamma,` generano le lettere greche. `\Gamma, \Delta,` generano le maiuscole. \in si chiama `\in`, \subset si chiama `\subset`, \rightarrow si chiama `\to`, ∞ si chiama `\infty`. Si veda la tabella allegata.

$x^\alpha \rightarrow +\infty$ si scrive perciò: `X^{\alpha}\to +\infty` (le graffe attorno ad α si possono anche omettere).

Simboli con indici: sommatoria \sum ed integrale \int si chiamano `\sum` e `\int`. Gli estremi si scrivono come se fossero pedici o esponenti:

$$\sum_{i=1}^{2n} \quad \text{genera:}$$

$$\int_0^{a^2} \quad \text{genera:}$$

Altri simboli con indici si scrivono nello stesso modo, ad esempio:

$$\lim_{t \rightarrow \infty} \quad \text{genera:}$$

$$\max_{0 < i < N} \quad \text{genera:}$$

Nomi di funzioni: nell'esempio sopra, `lim` e `max` vengono automaticamente in roman. Lo stesso avviene per `\sin`, `\cos`, `\exp`, `\log`, `\ln`, e diverse altre (v. tabella allegata), che se battute senza la `\` verrebbero scritte (erroneamente) in italic.

Qualche esempio:

$$\int_1^x \frac{dy}{y} = \log x$$

si scrive: `\int_1^x {dy\over y} = \log x`

$$\lim_{n \rightarrow \infty} \sum_{i=0}^n \frac{x^n}{n!} = e^x$$

si scrive: `\lim_{n \to \infty} \sum_{i=0}^n {x^n \over n!} = e^x`

$$X_{CM} = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}$$

si scrive: `X_{CM} = \frac {\sum_{i=1}^n m_i x_i} {\sum_{i=1}^n m_i}`.

Nota: a numeratore o denominatore di una formula in evidenza, oppure nelle formule inserite nel testo, `TeX` passa automaticamente dalla scrittura $\sum_{i=1}^n$ a $\sum_{i=1}^n$ per risparmiare spazio in verticale. In maniera analoga si comporta in varie altre situazioni (es. $\lim_{x \rightarrow \infty}$ e $\lim_{x \rightarrow \infty}$). E' comunque possibile forzarlo a seguire uno stile o l'altro (v. `\ds`, `\ts` per il `vanilla.sty`, oppure `\displaystyle`, `\textstyle` nel manuale "The `TeX`book").

Parentesi: tonde e quadre si eseguono normalmente, le graffe si scrivono `\{e\}` (per non confonderle con i caratteri di controllo); altre parentesi hanno un nome: ad esempio `\langle e \rangle` si scrivono `\langle e \rangle` e `\rangle`. E' allegata una lista.

Parentesi ed altri “delimitatori” (es. la barretta verticale `\vert`) sono disponibili più o meno ingranditi: `\big [`, `\Big [`, `\bigg [`, `\Bigg [`, generano quadre aperte di dimensione crescente. Una scrittura del tipo

`\left [(...formula a piacere ...) \right]`

genera [...] di dimensioni adattate a quello che sta in mezzo; `\left\{.....\right\}` genera {...} di dimensioni adattate.

Attenzione: il comando `\left` deve essere sempre appaiato con il comando `\right` e viceversa. Se occorre un solo delimitatore, chiudere od aprire con `\right.` e `\left.` rispettivamente, cioè mettere un “.” al posto del delimitatore mancante. Ad esempio:

$$x^2 = \sqrt{\frac{c+1}{d}}$$

si scrive:

\$\$

`x^2 = \left/ {c+1 \over d} \right.`

\$\$

Varie

$\sqrt{x+1}$ si scrive `\sqrt{x+1}`

(la dimensione della radice si adatta automaticamente al contenuto)

$\sqrt[10]{x+1}$ si scrive `\root 10 \of {x+1}`

x_1, \dots, x_n si scrive `x_1, \ldots, x_n`

- ★ `\ldots` = puntini in linea;
- ★ `\cdots` = puntini al centro come in $x_1 + \dots + x_n$;
- ★ `\cdot` = singolo punto come in $x \cdot y$.

Gli “accenti matematici” (barra, tilde, cappuccio, etc.) hanno ciascuno un nome, e precedono la lettera cui si riferiscono. Per esempio:

<code>\tilde x</code>	genera:	\tilde{x}
<code>\hat x</code>	genera:	\hat{x}
<code>\dot x</code>	genera:	\dot{x}
<code>\ddot x</code>	genera:	\ddot{x}
<code>\overline{abc}</code>	genera:	\overline{abc}
<code>\underline{abc}</code>	genera:	\underline{abc}

E’ allegata una tabella.

Attenzione a non confondere gli accenti matematici con gli accenti previsti per lettere nel testo. Inoltre non confondere `\underline` e `\overline` che funzionano solo in linguaggio matematico con `\underbar` e `\overbar` previsti per il testo ordinario.

Allineamento di formule:

$$\begin{aligned}x &= 1 \\ y &= 2n + 3 \\ \sin z &= \alpha\end{aligned}$$

(con i simboli = incolonnati) si scrive:

```
$$
\eqalign {
x &= 1 \cr
y &= 2n + 3 \cr
\sin z &= \alpha \cr
}
```

Funziona così: `\cr` separa le righe (l'ultimo può essere omissso); `&` segnala, in ogni riga, il carattere di allineamento. Non dimenticare l'ultima } !!

E' sempre utile, per evitare errori, andare effettivamente a capo ai `\cr`, e dare una forma grafica in cui le varie parti siano ben riconoscibili anche nell'editing del testo. Spazi bianchi (non righe bianche) si possono inserire a piacere. Allineamenti più complicati (su più simboli) si possono fare, per esempio per tabelle (si veda il manuale).

Il vanilla.sty prevede anche il formato:

```
$$
\align
riga di formula seguita da \\ per separare le righe
\endalign
$$
```

sempre utilizzando `&` per segnalare il carattere di allineamento.

Matrici: si fanno con la stessa logica degli allineamenti, usando il comando `\matrix`: ad esempio:

$$\begin{pmatrix} 1 & 2 & 3 \\ x_1 & x_2 & x_3 \\ \alpha & \alpha\beta & \alpha\beta\gamma \end{pmatrix}$$

si scrive:

```
$$
\left(
\matrix {1 & 2 & 3 \cr
x_1 & x_2 & x_3 \cr
\alpha & \alpha\beta & \alpha\beta\gamma \cr
}
\right)
$$
```

(la scomposizione del comando in righe è solo un suggerimento, per dare più chiarezza al testo; si può scrivere tutto di seguito).

Il vanilla.sty per le matrici prevede solo il formato `\matrix \endmatrix` (v. manuale PCTEX), al posto di `\matrix {.....}`.

Numerazione di formule: al termine di una formula, prima della chiusura con il \$\$, si può mettere \eqno(cosa qualsiasi), e “(cosa qualsiasi)” viene scritta a destra (\leqno per la numerazione a sinistra). Per esempio:

\$\$

$$x = 1 \text{ \eqno (5.2)}$$

\$\$

genera:

$$x = 1 \tag{5.2}$$

Oppure:

\$\$

$$\begin{array}{l} \text{\eqalign {x \&= 1 \cr$$

$$y \&= 12 \cr$$

$$z \&= 123 \cr } \text{\leqno [A3]}$$

\$\$

genera:

$$x = 1$$

[A3]

$$y = 12$$

$$z = 123$$

(si osservi che il numero è centrato).

E' possibile numerare separatamente le diverse righe, usando \eqalignno al posto di \eqalign (\leqalignno per la numerazione a sinistra): ad esempio

$$x = 1 \tag{1.a}$$

$$y = 12 \tag{1.b}$$

si scrive:

\$\$

$$\text{\eqalignno {x \&= 1 \& (1.a) \cr$$

$$y \&= 12 \& (1.b) \cr}}$$

\$\$

Ogni riga ha qui due simboli &: uno per l'incolonnamento, uno per il numero.

Il vanilla.sty prevede il comando \tag; per la numerazione a destra bisogna usare, all'inizio del testo, il comando \TagsOnRight (v. manuale PCT_EX).

3. Definizione di comandi propri (“macroistruzioni”)

E’ una cosa che, nelle sue forme più semplici, è importante saper fare fin dall’inizio, perchè abbrevia di molto la scrittura. I due casi semplici di uso più frequente sono:

1. Comandi che danno un nome a parti di testo ripetute.

Se, ad esempio, compare più volte una cosa complicata come

$$\sum_{i=1}^n x_i^2 \quad \backslash\mathrm{sum}_{\{i=1\}}^n x_i^2$$

le si dà un nome scrivendo (preferibilmente in testa):

`\def\somma {\sum_{i=1}^n x_i^2 }.`

Poi, scrivendo semplicemente `\somma`, si richiama la definizione scritta entro `{ }`. Ad esempio:

`$$`

`1 + \somma = 33`

`$$`

genera:

$$1 + \sum_{i=1}^n x_i^2 = 33$$

Il nome (in questo esempio `somma`) è scelto ad arbitrio, ma *obbligatoriamente contiene solo lettere*, non numeri od altri simboli.

Altri esempi:

`\def\bx{{\bf x}}` definisce `\bx` come **x** in grassetto. Attenzione: `\def\bx {\bf x}` produce un disastro! Tutto quello che segue `x` sarà anch’esso scritto in grassetto, perchè `\bf x` non è tra parentesi (le parentesi che compaiono sono quelle della definizione).

`\def\vettx {(x_1, \ldots, x_n)}` genera, ogni volta che si scrive `\vettx`, (x_1, \dots, x_n) .

2. Comandi con argomenti variabili (sono i più utili)

Dovendo scrivere più volte:

$$\sum_{i=1}^n x_i^2, \quad \sum_{i=1}^n y_i^2, \quad \dots$$

lasciando un simbolo variabile, si fa un’unica definizione:

`\def\somma#1{\sum_{i=1}^n #1_i^2},`

ove `#1` sta per “variabile numero uno” (ve ne possono stare fino a nove); scrivendo poi `\somma x`, `\somma y` si ottengono le due sommatorie scritte sopra, mentre

$$\backslash\mathrm{somma} \{(3x+2)\}, \quad \text{genera} \quad \sum_{i=1}^n (3x+2)_i^2.$$

Altro esempio: si definisce


```

\def\npla#1{#1_1, \ldots, #1_n},
dopodichè
\npla x genera  $x_1, \dots, x_n$ 
\npla {\hat x} genera  $\hat{x}_1, \dots, \hat{x}_n$ 

```

Esempio con due variabili: si definisce

```

\def\npla #1#2{#1_1, \ldots, #1_#2}
\npla xn genera  $x_1, \dots, x_n$ 
dopodichè
\npla x{32} genera  $x_1, \dots, x_{32}$ 
\npla {ax}{2r+1} genera  $ax_1, \dots, ax_{2r+1}$ 

```

Altro esempio: dopo aver definito

```

\def\der#1#2{{d#1\over d#2}}

```

(der sta per derivata); attenzione alle doppie parentesi!

```

\der fx genera  $\frac{df}{dx}$ 
\der {x^2}{x}=2x genera  $\frac{dx^2}{dx} = 2x$ 

```

Nota. Per un uso semplice delle macroistruzioni con argomenti variabili, eliminare nella definizione tutti gli spazi bianchi attorno a #1, etc. (al momento dell'uso invece si possono inserire spazi bianchi a piacere).

Suggerimento importante: se capita che alcune definizioni diventano di uso frequente, conviene tenerle in un file a parte; se il nome del file è, ad esempio, macro.tex, si richiamano in un qualsiasi scritto mettendo in testa \input macro. Se si usa il Professional Editor utilizzare la funzione F6 che può contenere macros sia di frasi che di files.

* * * * *

Nota generale sui comandi: cominciano sempre con \, e possono contenere solo lettere dell'alfabeto. Numeri ed altri simboli non ne fanno parte, con la sola eccezione dei comandi formati da un solo carattere (come nel caso degli accenti).

\bf lx genera **lx** e tutto quello che segue in grassetto;
 \bf x1 non vuole dire niente: x è interpretato come parte del comando \bfx, che non esiste

```

{\bf x1} genera solo x1 in grassetto
\bf\`e genera è in grassetto

```

Simboli del linguaggio matematico

Sono qui raccolti i simboli più comuni; per la lista completa si veda il manuale. Per stampare questa documentazione: `$ itex simbmam; $ dvilgs8 simbmam; $ las simbmam`
`laser4 simbmam (oppure laser1)`.

1. Lettere greche minuscole.

α	<code>\alpha</code>	ι	<code>\iota</code>	ϱ	<code>\varrho</code>
β	<code>\beta</code>	κ	<code>\kappa</code>	σ	<code>\sigma</code>
γ	<code>\gamma</code>	λ	<code>\lambda</code>	ς	<code>\varsigma</code>
δ	<code>\delta</code>	μ	<code>\mu</code>	τ	<code>\tau</code>
ϵ	<code>\epsilon</code>	ν	<code>\nu</code>	υ	<code>\upsilon</code>
ε	<code>\varepsilon</code>	ξ	<code>\xi</code>	ϕ	<code>\phi</code>
ζ	<code>\zeta</code>	\omicron	<code>\omicron</code>	φ	<code>\varphi</code>
η	<code>\eta</code>	π	<code>\pi</code>	χ	<code>\chi</code>
θ	<code>\theta</code>	ϖ	<code>\varpi</code>	ψ	<code>\psi</code>
ϑ	<code>\vartheta</code>	ρ	<code>\rho</code>	ω	<code>\omega</code>

2. Lettere greche maiuscole.

Γ	<code>\Gamma</code>	Ξ	<code>\Xi</code>	Φ	<code>\Phi</code>
Δ	<code>\Delta</code>	Π	<code>\Pi</code>	Ψ	<code>\Psi</code>
Θ	<code>\Theta</code>	Σ	<code>\Sigma</code>	Ω	<code>\Omega</code>
Λ	<code>\Lambda</code>	Υ	<code>\Upsilon</code>		

Le lettere greche maiuscole (a differenza delle minuscole) esistono anche in grassetto e in italic (più precisamente “math italic”); si richiamano con `\bf` e `\mit` rispettivamente. Il risultato è

$\Gamma \Delta \Theta \Lambda \Xi \Pi \Sigma \Upsilon \Phi \Psi \Omega$ $\Gamma \Delta \Theta \Lambda \Xi \Pi \Sigma \Upsilon \Phi \Psi \Omega$

3. Maiuscole “Calligrafiche”.

`$_{\cal A}$, $_{\cal B}$, ..., $_{\cal Z}$` generano $\mathcal{A}, \mathcal{B}, \dots, \mathcal{Z}$. Attenzione alle `{ }`: sono indispensabili per limitare il campo di `\cal` (come con `\bf`, `\it`...). L’alfabeto calligrafico completo è

$A B C D E F G H I J K L M N O P Q R S T U V W X Y Z$

Nel PC-TEX si usa `\Cal` al posto di `\cal`.

4. Simboli vari.

\aleph	<code>\aleph</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>
\hbar	<code>\hbar</code>	∇	<code>\nabla</code>	\forall	<code>\forall</code>
\Re	<code>\Re</code>	$'$	<code>\prime</code>	\spadesuit	<code>\spadesuit</code>
\Im	<code>\Im</code>	\parallel	<code>\parallel</code>	\diamondsuit	<code>\diamondsuit</code>
∂	<code>\partial</code>	\triangle	<code>\triangle</code>	\heartsuit	<code>\heartsuit</code>
∞	<code>\infty</code>	\backslash	<code>\backslash</code>	\clubsuit	<code>\clubsuit</code>

5. Operatori grandi, predisposti per avere indici.

Questi simboli compaiono in due misure: la più piccola nel testo, la più grande nelle formule in evidenza (si può però forzare la scelta: vedi `\textstyle` e `\displaystyle` nel manuale `TEX`).

Σ	\sum	<code>\sum</code>	\oint	<code>\oint</code>	\oplus	\bigoplus	<code>\bigoplus</code>
\prod	\prod	<code>\prod</code>	\bigcap	<code>\bigcap</code>	\otimes	\bigotimes	<code>\bigotimes</code>
\int	\int	<code>\int</code>	\bigcup	<code>\bigcup</code>			

6. Simboli di operazione binaria.

Oltre a $+$ e $-$ esistono

\pm	<code>\pm</code>	\circ	<code>\circ</code>	\vee	<code>\vee</code>
\mp	<code>\mp</code>	\bullet	<code>\bullet</code>	\wedge	<code>\wedge</code>
\setminus	<code>\setminus</code>	\div	<code>\div</code>	\oplus	<code>\oplus</code>
\cdot	<code>\cdot</code>	\cap	<code>\cap</code>	\otimes	<code>\otimes</code>
\times	<code>\times</code>	\cup	<code>\cup</code>	\odot	<code>\odot</code>
\ast	<code>\ast</code>	\bigcirc	<code>\bigcirc</code>	\dagger	<code>\dagger</code>
\star	<code>\star</code>	\bigtriangleup	<code>\bigtriangleup</code>	\ddagger	<code>\ddagger</code>
\diamond	<code>\diamond</code>	\bigtriangledown	<code>\bigtriangledown</code>		

attorno a questi simboli è previsto un piccolo spazio: perciò, ad esempio, $A \setminus B$ e $A \setminusminus B$ sono diversi, e danno rispettivamente $A \setminus B$ e $A \setminusminus B$.

7. Relazioni.

Oltre a $>$, $<$ e $=$ esistono

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\sim	<code>\sim</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\simeq	<code>\simeq</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\approx	<code>\approx</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\cong	<code>\cong</code>
$ $	<code>\mid</code>	\parallel	<code>\parallel</code>	\doteq	<code>\doteq</code>
\propto	<code>\propto</code>	\perp	<code>\perp</code>	\mapsto	<code>\mapsto</code>
\leftarrow	<code>\leftarrow</code>	\rightarrow	<code>\rightarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>

Le relazioni “negate” si ottengono premettendo `\not`: ad esempio `\not=` e `\not\equiv` generano \neq e \nequiv . `\not\in` genera \notin , che (secondo Knuth) non è molto bello; al suo posto si può usare `\notin`, che genera \notin (secondo Knuth è molto meglio).

Per le frecce esistono versioni un pò più lunghe, di nome `\longmapsto`, `\longleftarrow`, `\Longleftarrow`... che generano \longmapsto , \longleftarrow , \Longleftarrow ...

Il comando `\buildrel<scritta>\over<relazione>` (simile alle frazioni) permette di scrivere

qualche cosa sopra un qualsiasi simbolo di relazione. Ad esempio:

$\stackrel{\text{def}}{=}$	<code>\buildrel\rm def\over=</code>
$\xrightarrow{t \rightarrow \infty}$	<code>\buildrel t\rightarrow\infty\over\longrightarrow</code>

Attorno alle relazioni è previsto un leggero spazio, perciò, ad esempio, i simboli `\mid` e `\parallel` non sono adatti per i simboli di modulo e norma, e vanno preferiti i corrispondenti “delimitatori” (v. oltre). Si ottiene $|v|$ con `\mid`, e $\|v\|$ con `\|`. Il simbolo `\mid` è più adatto a scritture del tipo $\{v \in \mathbf{R}^n \mid u \cdot v \leq 1\}$.

8. Accenti matematici.

\hat{x}	<code>\hat x</code>	\tilde{x}	<code>\tilde x</code>	\dot{x}	<code>\dot x</code>
\ddot{x}	<code>\ddot x</code>	\bar{x}	<code>\bar x</code>	\vec{x}	<code>\xvec x</code>

9. Parentesi e altri delimitatori.

$($	<code>(</code>	$)$	<code>)</code>
$[$	<code>[o \lbrack</code>	$]$	<code>] o \rbrack</code>
$\{$	<code>\{ o \lbrace</code>	$\}$	<code>\} o \rbrace</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>
$ $	<code> o \vert</code>	$\ $	<code>\ o \Vert</code>

Per tutti esistono le versioni più grandi, che si ottengono premettendo `\big`, `\Big`, `\bigg`, `\Bigg`. Con la parentesi quadra e tonda il risultato è

$$[([([([([(()]])])])])]$$

10. Spaziatura fine nelle formule.

Oltre a `\(spazio)`, `\quad` e `\qquad` esistono

<code>\,</code>	spazio sottile, circa 1/6 di <code>\quad</code>
<code>\></code>	spazio medio, circa 2/9 di <code>\quad</code>
<code>\;</code>	spazio grosso, circa 5/18 di <code>\quad</code>
<code>\!</code>	spazio sottile <i>negativo</i> , circa 1/6 di <code>\quad</code>

