

Corso \LaTeX

\LaTeX e la matematica

Riccardo Dossena

Liceo Scientifico "G. Novello" - Codogno (LO)

28 ottobre 2019



Per la matematica in \LaTeX è quasi indispensabile caricare i pacchetti \mathcal{AMS}

```
\documentclass [11pt,a4paper]{article}

\usepackage [T1]{fontenc}
\usepackage [utf8]{inputenc}
\usepackage [italian]{babel}
\usepackage{lmodern}

\usepackage{amsmath,amssymb,amsthm} % pacchetti AmS utili
                                     % per la matematica
```

Da ora in avanti supporremo che i pacchetti \mathcal{AMS} siano caricati

Modalità matematica – in linea e in display

Le formule matematiche possono essere “in linea” o “in display”

Questa qui $x^n + y^n = z^n$ è una formula in linea, cioè nel testo.

Questa qui $x^n + y^n = z^n$ è una formula in linea, cioè nel testo.

Questa qui invece è una formula in display

```
\[  
y = \frac{\sqrt{x+1}}{x}
```

```
\]  
e questa è una formula numerata
```

```
\begin{equation}  
f(x) = \sqrt[3]{x-2}
```

```
\end{equation}
```

Questa qui invece è una formula in display

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

e questa è una formula numerata

$$f(x) = \sqrt[3]{x-2} \quad (1)$$

Gli spazi non hanno alcun effetto in modalità matematica, poiché la spaziatura è gestita automaticamente da \LaTeX

Scrivere $x=1$ oppure

```
\[  
x+y=(1+z)
```

```
\]  
dà lo stesso risultato di  $x = 1$  o
```

```
\[  
x + y = (1 + z)
```

```
\]
```

Scrivere $x = 1$ oppure

$$x + y = (1 + z)$$

dà lo stesso risultato di $x = 1$ o

$$x + y = (1 + z)$$



Non si deve *mai* lasciare alcuna linea bianca nell'ambiente matematico "in display", altrimenti si riceve un messaggio di errore!

Le formule in linea e in display non appaiono allo stesso modo: i comandi `\displaystyle` e `\textstyle` consentono di intercambiare i due stili

Le formule in linea, come `$$\frac{x+1}{x}$$`, appaiono spesso diverse dalle stesse in display:

```
\[
  \frac{x+1}{x}
\]
```

Se vogliamo in linea lo stesso stile usato in display dobbiamo usare il comando `\verb|\displaystyle|` e scrivere `$$\displaystyle{\frac{x+1}{x}}$`. `\LaTeX` cercherà di aggiustare nel migliore dei modi l'interlinea.

Viceversa, il comando `\verb|\textstyle|` consente di ottenere in display lo stile in linea:

```
\[
  \textstyle{\frac{x+1}{x}}
\]
```

Le formule in linea, come $\frac{x+1}{x}$, appaiono spesso diverse dalle stesse in display:

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

Se vogliamo in linea lo stesso stile usato in display dobbiamo usare il comando `\displaystyle` e scrivere `\frac{x+1}{x}`. `\LaTeX` cercherà di aggiustare nel migliore dei modi l'interlinea.

Viceversa, il comando `\textstyle` consente di ottenere in display lo stile in linea:

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

Per default, le formule in display sono centrate. Se si vogliono allineare a sinistra bisogna usare l'opzione di classe `fleqn`

```
\documentclass[11pt,a4paper,fleqn]{article}
...
\begin{document}
Con questa opzione di classe le formule
vengono allineate a sinistra, come nel
caso di
\[
y=2x-3
\]
oppure come
\begin{equation}
y=\frac{x^2-1}{x+2}
\end{equation}
\end{document}
```

Con questa opzione di classe le formule vengono allineate a sinistra, come nel caso di

$$y = 2x - 3$$

oppure come

$$y = \frac{x^2 - 1}{x + 2} \tag{1}$$

I riferimenti incrociati con le formule si ottengono coi comandi `\label` e `\eqref`

```
\chapter{Riferimenti e formule}
```

```
\section{La formula di Eulero}
```

Da un'indagine effettuata fra i matematici è emerso che la formula più bella in assoluto della matematica è la relazione di Eulero. Si tratta dell'uguaglianza

```
\begin{equation}\label{eq:eulero}
```

$$e^{i\pi} + 1 = 0.$$

```
\end{equation}
```

La formula-`\eqref{eq:eulero}` contiene i cinque numeri fondamentali dell'analisi:

```
$e$, $i$, $\pi$, $1$ e~$0$.
```

Capitolo 1

Riferimenti e formule

1.1 La formula di Eulero

Da un'indagine effettuata fra i matematici è emerso che la formula più bella in assoluto della matematica è la relazione di Eulero. Si tratta dell'uguaglianza

$$e^{i\pi} + 1 = 0. \tag{1.1}$$

La formula (1.1) contiene i cinque numeri fondamentali dell'analisi: e , i , π , 1 e 0 .

Se si usa `\ref` anziché `\eqref` non vengono stampate automaticamente le parentesi

L^AT_EX mette in corsivo tutte le lettere inserite in modalità matematica

Per inserire brevi testi si usa il comando `\text`, specificando la spaziatura prima e dopo

Consideriamo la seguente funzione

```
\[  
  f(x)=x^3 \quad \text{per } x<0$  
\]
```

Consideriamo la seguente funzione

$$f(x) = x^3 \quad \text{per } x < 0$$

Il comando `\quad` (“quadrato”) inserisce uno spazio di `1em`; non serve in linea:

La funzione `$f(x)=x^3$` per `$x<0$`

La funzione $f(x) = x^3$ per $x < 0$

Esponenti e indici

Gli esponenti si ottengono con `^` e gli indici con `_`

```
\[  
x^2+y^2=1 \quad x_1=x_2 \quad x_1^2=x_2^2  
\]
```

$$x^2 + y^2 = 1 \quad x_1 = x_2 \quad x_1^2 = x_2^2$$

Se l'indice o l'esponente sono formati da più di un carattere è necessario usare delle graffe di raggruppamento

```
\[  
x^{2n}+y^{3m}=1 \quad a_{ij}=3^{i+j} \quad x^{y^2}=a_{i_n}  
\]
```

$$x^{2n} + y^{3m} = 1 \quad a_{ij} = 3^{i+j} \quad x^{y^2} = a_{i_n}$$

`\quad` (“quadratone”) inserisce uno spazio pari a due `\quad`

Le graffe si usano per formare il gruppo a cui va applicato l'esponente

`\noindent` C'è una lieve differenza fra

```
\[
(x^2+y^2)^n \quad \text{e}
\quad \{(x^2+y^2)\}^n
```

`\]`
Nel primo caso l'esponente è applicato solo alla parentesi `)`, nel secondo a tutto il gruppo `(x^2+y^2)`. Entrambe le espressioni sono corrette, anche se Knuth preferisce la prima perché più facile da editare e da leggere.

C'è una lieve differenza fra

$$(x^2 + y^2)^n \quad \text{e} \quad (x^2 + y^2)^n$$

Nel primo caso l'esponente è applicato solo alla parentesi `)`, nel secondo a tutto il gruppo `(x^2 + y^2)`. Entrambe le espressioni sono corrette, anche se Knuth preferisce la prima perché più facile da editare e da leggere.

Le frazioni si compongono col comando

$$\backslash\text{frac}\{\langle\text{numeratore}\rangle\}\{\langle\text{denominatore}\rangle\}$$

Una frazione in linea appare più piccola, come `\frac{1}{2}`, mentre una in display appare come

```
\[  
  \frac{1}{x+2}  
\]
```

Se si vuole una frazione in linea come appare in display basta usare `\verb|\dfrac|`. Ad esempio si può ottenere in linea la frazione `\dfrac{1}{2+y}` proprio come in display. Il risultato non è molto bello, per cui se è proprio necessario forse è meglio scrivere `1/(2+y)`.

Una frazione in linea appare più piccola, come $\frac{1}{2}$, mentre una in display appare come

$$\frac{1}{x+2}$$

Se si vuole una frazione in linea come appare in display basta usare `\dfrac`. Ad esempio si può ottenere in linea la frazione $\frac{1}{2+y}$ proprio come in display. Il risultato non è molto bello, per cui se è proprio necessario forse è meglio scrivere $1/(2+y)$.

Nome	Descrizione	Esempio
<code>\quad</code>	<i>quadrato</i> , pari a $1em$	□
<code>\qquad</code>	<i>quadrato</i> , pari a $2em$	□□
<code>\,</code>	<i>spazio fine</i> , pari a $\frac{3}{18}em$	∥
<code>\:</code>	<i>spazio medio</i> , pari a $\frac{4}{18}em$	∥
<code>\;</code>	<i>spazio grande</i> , pari a $\frac{5}{18}em$	∥
<code>\enskip</code>	pari a $\frac{1}{2}em$	□
<code>\!</code>	<i>spazio negativo</i> , pari a $-\frac{3}{18}em$	∥

Nota: questi comandi funzionano anche in modalità testo, ma in ambiente non matematico si consiglia di limitare il loro uso a casi particolari

Somme e prodotti

I simboli di somma e prodotto si ottengono rispettivamente coi comandi `\sum` e `\prod`

I simboli di somma e prodotto si possono usare sia in display

```
\[
  \sum_{i=1}^n a_i = a_1+\ldots+a_n
\]
\[
  \prod_{i=1}^n x_i =
  x_1\cdot \ldots \cdot x_n
\]
```

che in linea, dove appaiono in modo diverso ma più adatto

`\sum_{n=0}^{+\infty} a_n`, con gli estremi affiancati.

I simboli di somma e prodotto si possono usare sia in display

$$\sum_{i=1}^n a_i = a_1 + \dots + a_n$$

$$\prod_{i=1}^n x_i = x_1 \cdot \dots \cdot x_n$$

che in linea, dove appaiono in modo diverso ma più adatto $\sum_{n=0}^{+\infty} a_n$, con gli estremi affiancati.

I limiti si possono ottenere col comando `\lim`

```
\noindent Calcoliamo i seguenti limiti  
\[  
  \lim_{x\to+\infty} \frac{x}{x+1}=1  
  \quad \lim_{x\to 3^+} (x-3)=0^+  
\]
```

mentre in linea si otterrebbe qualcosa come `$$\lim_{x\to x_0} f(x)=f(x_0)$$` (almeno per le funzioni continue).

Un'altra possibilità è scrivere

```
\[  
  f(x) \to L \quad \text{per } x\to x_0$.  
\]
```

Calcoliamo i seguenti limiti

$$\lim_{x \rightarrow +\infty} \frac{x}{x+1} = 1 \quad \lim_{x \rightarrow 3^+} (x-3) = 0^+$$

mentre in linea si otterrebbe qualcosa come $\lim_{x \rightarrow x_0} f(x) = f(x_0)$ (almeno per le funzioni continue).

Un'altra possibilità è scrivere

$$f(x) \rightarrow L \quad \text{per } x \rightarrow x_0.$$

Limiti con frecce old-arrows

Il pacchetto `old-arrows` (eventualmente con l'opzione `new`) consente di ottenere frecce con punte più piccole

$$\lim_{x \rightarrow x_0} f(x) \quad \text{vs.} \quad \lim_{x \rightarrow x_0} f(x)$$

default old-arrows

```
...
\usepackage[new]{old-arrows}
...
Con il pacchetto \texttt{old-arrows}
si può ottenere
\[
\lim_{x \to x_0} f(x) \quad \text{e} \quad f(x) \to L
\]
anziché
\[
\lim_{x \vartto x_0} f(x) \quad \text{e} \quad f(x) \vartto L
\]
```

Con il pacchetto `old-arrows` si può ottenere

$$\lim_{x \rightarrow x_0} f(x) \quad \text{e} \quad f(x) \rightarrow L$$

anziché

$$\lim_{x \rightarrow x_0} f(x) \quad \text{e} \quad f(x) \rightarrow L$$

Le derivate si scrivono con il simbolo ' (apostrofo)

```
\noindent Le derivate si scrivono così
```

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

```
oppure così
```

```
\[
  f(x)=x^2 \quad f'(x)=2x
  \quad f''(x)=2
\]
```

```
oppure ancora così
```

```
\[
  y=x^2 \quad \frac{dy}{dx}=2x
  \quad \frac{d^2 y}{dx^2}=2
\]
```

Le derivate si scrivono così

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

oppure così

$$f(x) = x^2 \quad f'(x) = 2x \quad f''(x) = 2$$

oppure ancora così

$$y = x^2 \quad \frac{dy}{dx} = 2x \quad \frac{d^2 y}{dx^2} = 2$$

Integrali – Il pacchetto esint

Col pacchetto **esint** si possono ottenere altri tipi di integrali, oltre che un miglioramento di quelli di default

```
...
\usepackage{esint}
...
Gli integrali di \texttt{esint} vengono
ridefiniti e diventano, ad esempio
\l[
  \iint_D f(x,y)\,dx\,dy \quad \iint_D f(x,y)\,dx\,dy
  \oiint f(z)\,dz
\]
e in aggiunta
\l[
  \oiint_D f(x,y)\,dx\,dy \quad \oiint_D f(x,y)\,dx\,dy
  \oiintclockwise f(z)\,dz
\]
notare le differenze con quelli di
default.
```

Gli integrali di **esint** vengono ridefiniti e diventano, ad esempio

$$\iint_D f(x,y) dx dy \quad \oiint f(z) dz$$

e in aggiunta

$$\oiint_D f(x,y) dx dy \quad \oiintclockwise f(z) dz$$

notare le differenze con quelli di default.

Operatori funzionali

I nomi delle funzioni che \LaTeX mette a disposizione sono quelle ordinarie e corrispondenti alle prescrizioni UNI-ISO

\LaTeX li classifica come *operatori* e imposta automaticamente la spaziatura corretta a destra e a sinistra

possono avere o non avere limiti

Gli operatori funzionali senza limiti si scrivono così

```
\[  
  \sin x \quad \sin^2 x  
  \quad \log_2 (x+1)  
\]
```

e gli esponenti e gli indici appaiono normalmente.

Negli operatori con limiti, se vengono applicati esponenti e indici, questi appaiono in display sopra e sotto

```
\[  
  \lim f(x) \quad \lim_{x \to 2} g(x)  
  \quad \sup_{x \in A} h(x)  
\]
```

Gli operatori funzionali senza limiti si scrivono così

$$\sin x \quad \sin^2 x \quad \log_2(x+1)$$

e gli esponenti e gli indici appaiono normalmente.

Negli operatori con limiti, se vengono applicati esponenti e indici, questi appaiono in display sopra e sotto

$$\lim f(x) \quad \lim_{x \rightarrow 2} g(x) \quad \sup_{x \in A} h(x)$$

Operatori funzionali e nuovi operatori

Operatori senza limiti

<code>\arccos</code>	<code>\cosh</code>	<code>\dim</code>	<code>\ln</code>	<code>\tan</code>
<code>\arcsin</code>	<code>\cot</code>	<code>\exp</code>	<code>\log</code>	<code>\tanh</code>
<code>\arctan</code>	<code>\coth</code>	<code>\hom</code>	<code>\sec</code>	
<code>\arg</code>	<code>\csc</code>	<code>\ker</code>	<code>\sin</code>	
<code>\cos</code>	<code>\deg</code>	<code>\lg</code>	<code>\sinh</code>	

Operatori con limiti

<code>\det</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>	<code>\Pr</code>
<code>\gcd</code>	<code>\lim</code>	<code>\limsup</code>	<code>\min</code>	<code>\sup</code>

È possibile definire nuovi operatori (senza e con limiti) tramite i comandi `\DeclareMathOperator` e `\DeclareMathOperator*` da inserire nel preambolo

```
\DeclareMathOperator{<comando>}{<nome-operatore>} % senza limiti  
\DeclareMathOperator*{<comando>}{<nome-operatore>} % con limiti
```

```

...
\DeclareMathOperator{\sen}{sen}
\DeclareMathOperator*{\maxlim}{max\,lim}
...
\begin{document}
Una volta si usava scrivere
\[
\sen x
\]
ma adesso si preferisce la notazione
\[
\sin x
\]
anche perché in conformità con le
norme-UNI-ISO.

Una possibile nuova definizione di un
operatore con limiti è la seguente
\[
\maxlim_{x\to+\infty} f(x)
\]
dove possiamo notare anche lo spazio
fra le due parti del nome.
\end{document}

```

Una volta si usava scrivere

$$\sen x$$

ma adesso si preferisce la notazione

$$\sin x$$

anche perché in conformità con le
norme UNI-ISO.

Una possibile nuova definizione di
un operatore con limiti è la seguente

$$\maxlim_{x \rightarrow +\infty} f(x)$$

dove possiamo notare anche lo spazio
fra le due parti del nome.

Coefficienti binomiali

I coefficienti binomiali si ottengono col comando

```
\binom{⟨1° termine⟩}{⟨2° termine⟩}
```

In linea i coefficienti binomiali
appaiono così `\binom{4}{3}`
mentre in display

```
\[  
  \binom{n}{k}=\frac{n!}{k!(n-k)!}  
\]
```

In linea i coefficienti binomiali
appaiono così $\binom{4}{3}$ mentre in display

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Simboli insiemistici

\subset	<code>\subset</code>	\supset	<code>\supset</code>	\cup	<code>\cup</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cap	<code>\cap</code>
\in	<code>\in</code>	\ni o \owns	<code>\ni</code> o <code>\owns</code>	\notin	<code>\notin</code>
\complement	<code>\complement</code>	\setminus	<code>\setminus</code>	$ $	<code>\mid</code>
\emptyset	<code>\emptyset</code>	\emptyset	<code>\varnothing</code>		

Operatori insiemistici

\bigcup	<code>\bigcup</code>	\bigcap	<code>\bigcap</code>
-----------	----------------------	-----------	----------------------

Per indicare che l'elemento x appartiene all'insieme A si scrive

```
\[
  x\in A \quad\text{oppure}\quad A\ni x.
\]
```

L'insieme degli elementi di A che soddisfano la proprietà P si scrive

```
\[
  \{x\in A \mid P(x)\} \quad\text{oppure}\quad \text{quad}
  \{x\in A : P(x)\}.
\]
```

Per indicare che A è un *sottoinsieme* di B si scrive

```
\[
  A\subset B \quad\text{oppure}\quad B\supset A
\]
```

e se si ammette che A e B possano essere anche uguali

```
\[
  A\subseteq B \quad\text{oppure}\quad \text{quad}
  B\supseteq A.
\]
```

Si ha che

```
\[
  A\cap B = \{x\mid x\in A \text{ e } x\in B\}
\]
```

e inoltre

```
\[
  \bigcap_{i=1}^n A_i = A_1 \cap \dots \cap A_n.
\]
```

Per indicare che l'elemento x appartiene all'insieme A si scrive

$$x \in A \quad \text{oppure} \quad A \ni x.$$

L'insieme degli elementi di A che soddisfano la proprietà P si scrive

$$\{x \in A \mid P(x)\} \quad \text{oppure} \quad \{x \in A : P(x)\}.$$

Per indicare che A è un *sottoinsieme* di B si scrive

$$A \subset B \quad \text{oppure} \quad B \supset A$$

e se si ammette che A e B possano essere anche uguali

$$A \subseteq B \quad \text{oppure} \quad B \supseteq A.$$

Si ha che

$$A \cap B = \{x \mid x \in A \text{ e } x \in B\}$$

e inoltre

$$\bigcap_{i=1}^n A_i = A_1 \cap \dots \cap A_n.$$

Stile	Codice	Risultato
Tondo	<code>\mathrm{ABCdef123}</code>	ABCdef123
Corsivo	<code>\mathit{ABCdef123}</code>	<i>ABCdef123</i>
Grassetto	<code>\mathbf{ABCdef123}</code>	ABCdef123
Macchina da scrivere	<code>\mathtt{ABCdef123}</code>	ABCdef123
Senza grazie	<code>\mathsf{ABCdef123}</code>	ABCdef123
Gotico	<code>\mathfrak{ABCdef123}</code>	<mathfrak{abcdef123}< math=""></mathfrak{abcdef123}<>
Blackboard Bold	<code>\mathbb{ABC}</code>	\mathbb{ABC}
Calligrafico	<code>\mathcal{ABC}</code>	\mathcal{ABC}
Manoscritto [†]	<code>\mathscr{ABC}</code>	\mathscr{ABC}

[†]Richiede il pacchetto `mathrsfs`

Gli insiemi numerici si ottengono con `\mathbb`. Conviene definire dei comandi per semplificare la scrittura

```
...
\newcommand{\N}{\mathbb{N}} % nuovi comandi
\newcommand{\Z}{\mathbb{Z}} % per gli insiemi
\newcommand{\Q}{\mathbb{Q}} % numerici
\newcommand{\R}{\mathbb{R}}
\newcommand{\C}{\mathbb{C}}

\newcommand{\U}{\mathcal{U}}
...
\begin{document}
Lo stile Blackboard Bold dà
\[
\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
]
Gli insiemi numerici sono tali che
\[
\N\subset\Z\subset\Q\subset\R\subset\C.
]
Scrivere  $\mathbb{R}$  dà lo stesso risultato di
 $\mathbb{R}$ .

Le famiglie di insiemi si possono indicare
con il carattere calligrafico. Ad esempio
 $\mathcal{U}$  può essere una topologia
su  $\mathbb{R}$ , dunque  $\varnothing \in \mathcal{U}$  e
 $\mathbb{R} \in \mathcal{U}$ .

 $\mathcal{U}$  dà lo stesso risultato di  $\mathcal{U}$ .
\end{document}
```

Lo stile Blackboard Bold dà

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Gli insiemi numerici sono tali che

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}.$$

Scrivere \mathbb{R} dà lo stesso risultato di \mathbb{R} .

Le famiglie di insiemi si possono indicare con il carattere calligrafico. Ad esempio \mathcal{U} può essere una *topologia* su \mathbb{R} , dunque $\varnothing \in \mathcal{U}$ e $\mathbb{R} \in \mathcal{U}$.

\mathcal{U} dà lo stesso risultato di \mathcal{U} .

Delimitatori

Sono disponibili molti delimitatori come parentesi e barre verticali, di dimensioni diverse e adattabili alle situazioni

Delimitatore	Codice	Delimitatore	Codice
(<code>(</code>)	<code>)</code>
[<code>[</code>]	<code>]</code>
{	<code>\{</code>	}	<code>\}</code>
	<code>\lvert</code>		<code>\rvert</code>
	<code>\lVert</code>		<code>\rVert</code>
⌊	<code>\lfloor</code>	⌋	<code>\rfloor</code>
⌈	<code>\lceil</code>	⌉	<code>\rceil</code>
⟨	<code>\langle</code>	⟩	<code>\rangle</code>

I delimitatori si possono estendere verticalmente in modo automatico antepoendo `\left` e `\right`

Possiamo usare le parentesi normalmente

```
\[  
  (x+y)^2
```

```
\]  
oppure estendere i delimitatori
```

```
\[  
  \left(\frac{1}{x}+\frac{1}{y}\right)^2
```

```
\]
```

Possiamo usare le parentesi normalmente

$$(x + y)^2$$

oppure estendere i delimitatori

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

Se vogliamo un delimitatore esteso solo destro o sinistro, dobbiamo usare dall'altra parte i *delimitatori vuoti* `\left.` o `\right.`.

Se per qualche motivo volessimo scrivere

```
\[  
  \left.\{\frac{x}{y}\right.  
  \quad\text{oppure}\quad  
  \left.\sum x_i \right]
```

```
\]
```

Se per qualche motivo volessimo scrivere

$$\left.\left\{\frac{x}{y}\right.\right. \quad \text{oppure} \quad \left.\sum x_i\right]$$

Ci sono però dei casi in cui il ridimensionamento automatico fornito da `\left` e `\right` non dà risultati soddisfacenti

Sono disponibili allora i comandi seguenti, da anteporre ai delimitatori, che ingrandiscono in modo fisso

Comandi per delimitatori sinistri

Codice	<code>\bigl(</code>	<code>\Bigl(</code>	<code>\biggl(</code>	<code>\Biggl(</code>
Risultato	((((

Comandi per delimitatori destri

Codice	<code>\bigr)</code>	<code>\Bigr)</code>	<code>\biggr)</code>	<code>\Biggr)</code>
Risultato))))

Con questi comandi non è necessario l'uso di delimitatori vuoti

Confrontiamo

$$\left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p}$$

contro

$$\left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p}$$

ottenuti rispettivamente con

```
\left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p}
```

e

```
\biggl[\sum_i a_i \Bigl| \sum_j x_{ij} \Bigr|^p \biggr]^{1/p}
```

Nel secondo caso la resa tipografica è migliore

oppure

$$((a_1b_1) - (a_2b_2)) ((a_2b_1) + (a_1b_2))$$

contro

$$((a_1b_1) - (a_2b_2))((a_2b_1) + (a_1b_2))$$

ottenuti rispettivamente con

```
\left((a_1 b_1) - (a_2 b_2)\right) \left((a_2 b_1) + (a_1 b_2)\right)
```

e

```
\bigl((a_1 b_1) - (a_2 b_2)\bigr) \bigl((a_2 b_1) + (a_1 b_2)\bigr)
```

Nel secondo caso l'annidamento è visibile in modo migliore

Valore assoluto – barra verticale

Il valore assoluto si ottiene in modo corretto con la coppia di delimitatori `\lvert` e `\rvert` e non con il simbolo `|`

Si ottiene lo stesso risultato se scriviamo

```
\[
|x| \quad \text{oppure} \quad \quad
\lvert x\rvert
```

ma bisogna fare attenzione, perché ad esempio in

```
\[
|-1|=1 \quad \text{e} \quad \quad
\lvert -1\rvert=1
```

la spaziatura nel primo modulo non è corretta.

Si ottiene lo stesso risultato se scriviamo

$|x|$ oppure $|x|$

ma bisogna fare attenzione, perché ad esempio in

$|-1|=1$ e $|-1|=1$

la spaziatura nel primo modulo non è corretta.

\LaTeX imposta correttamente le spaziature perché definisce `\lvert` e `\rvert` come *delimitatori*

Valore assoluto – barra verticale

Tuttavia, dopo un comando come `\left` e `\right` o `\big` e affini, al posto di `\lvert` o `\rvert` si può scrivere semplicemente `|`

Si ottiene lo stesso risultato se scriviamo

```
\[
\Bigl\lvert \frac{x-1}{x} \Bigr\rvert
\]
```

oppure se inseriamo da tastiera il simbolo della barra verticale

```
\[
\Bigl| \frac{x-1}{x} \Bigr|
\]
```

Si ottiene lo stesso risultato se scriviamo

$$\left| \frac{x-1}{x} \right|$$

oppure se inseriamo da tastiera il simbolo della barra verticale

$$\left| \frac{x-1}{x} \right|$$

Lo stesso si può fare con `\langle` e `\rangle`, scrivendo al loro posto `<` e `>`

Valore assoluto e norma – definizione dei comandi

Per semplificare l'uso dei valori assoluti e delle norme viene raccomandato di definire opportuni comandi

```
\newcommand{\abs}[1]{\lvert#1\rvert}
\newcommand{\norm}[1]{\lVert#1\rVert}
```

```
...
\newcommand{\abs}[1]{\lvert#1\rvert}
\newcommand{\norm}[1]{\lVert#1\rVert}
...
Una proprietà del valore assoluto è
\l[
  \abs{xy}=\abs{x}\abs{y}
\]
mentre per la norma
\l[
  \norm{\mathbf{x}}=\sqrt{x_1^2+x_2^2}
\]
Purtroppo non sono scalabili
\l[
  \abs{\frac{x}{y}} \quad \text{e} \quad \norm{\frac{1}{2}\mathbf{x}}
\]
\]
```

Una proprietà del valore assoluto è

$$|xy| = |x||y|$$

mentre per la norma

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2}$$

Purtroppo non sono scalabili

$$\left|\frac{x}{y}\right| \quad \text{e} \quad \left\|\frac{1}{2}\mathbf{x}\right\|$$

Valore assoluto – Il pacchetto mathtools

Caricando il pacchetto `mathtools` è possibile ottenere una definizione di valore assoluto (ed eventualmente di norma) ancora più conveniente inserendo nel preambolo

```
\DeclarePairedDelimiter\abs{\lvert}{\rvert}
```

```
...
\usepackage{mathtools}
\DeclarePairedDelimiter\abs{\lvert}{\rvert}
...
Il comando \verb|abs| funziona come prima
\l
  \abs{xy}=\abs{x}\abs{y}
\r
ma è possibile specificare un'opzione per la
dimensione voluta
\l
  \abs[\big]{\frac{x}{y}}
\r
oppure con la versione asteriscata diventa
scalabile come con \verb|\left| e \verb|\right|
\l
  \abs*{\frac{x}{y}}
\r
```

Il comando `\abs` funziona come prima

$$|xy| = |x||y|$$

ma è possibile specificare un'opzione per la dimensione voluta

$$\left| \frac{x}{y} \right|$$

oppure con la versione asteriscata diventa scalabile come con `\left` e `\right`

$$\left| \frac{x}{y} \right|$$

Uso della barra verticale (insiemi e restrizioni)

È importante rimarcare che la barra verticale nella notazione insiemistica viene generata dal comando `\mid`

```
\[
  \{ x\in\mathbb{R} \mid x < 1 \}
\]
```

$$\{x \in \mathbb{R} \mid x < 1\}$$

Se è necessario ridimensionare i delimitatori, per la barra verticale si usano i comandi `\bigm|`, `\Bigm|`, `\biggm|`, `\Biggm|`

```
\[
  \biggl\{ \frac{1}{n+1} \biggm|
  n\in\mathbb{N} \biggr\}
\]
```

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

La barra verticale si può usare anche nella restrizione di funzioni

```
\[
  f|_A(x), \enskip x\in A
\]
```

$$f|_A(x), \quad x \in A$$

Esercizio 18.14 – da D. E. Kunth, *The T_EXbook*, pag. 171

Alcuni matematici perversi usano le parentesi quadre rovesciate per denotare gli “intervalli aperti”. Spiegare come editare la seguente formula bizzarra: $]-\infty, T[\times]-\infty, T[.$

Esercizio 18.14 – da D. E. Kunth, *The T_EXbook*, pag. 171

Alcuni matematici perversi usano le parentesi quadre rovesciate per denotare gli “intervalli aperti”. Spiegare come editare la seguente formula bizzarra: $] - \infty, T[\times] - \infty, T[$.

Sarebbe errato scrivere semplicemente

```
\[  
  ]-\infty,T[ \times ]-\infty,T[  
\]
```

$$] - \infty, T[\times] - \infty, T[$$

perché la spaziatura non è corretta!

Esercizio 18.14 – da D. E. Kunth, *The T_EXbook*, pag. 171

Alcuni matematici perversi usano le parentesi quadre rovesciate per denotare gli “intervalli aperti”. Spiegare come editare la seguente formula bizzarra: $]-\infty, T[\times]-\infty, T[$.

Soluzione 1

```
\[  
  \left]-\infty,T\right[ \times \left]-\infty,T\right[  
\]
```

$$]-\infty, T[\times]-\infty, T[$$

Esercizio 18.14 – da D. E. Kunth, *The T_EXbook*, pag. 171

Alcuni matematici perversi usano le parentesi quadre rovesciate per denotare gli “intervalli aperti”. Spiegare come editare la seguente formula bizzarra: $] - \infty, T[\times] - \infty, T[$.

Soluzione 2

```
\[  
  \mathopen] - \infty, T \mathclose[ \times  
  \mathopen] - \infty, T \mathclose[  
\]
```

$$] - \infty, T[\times] - \infty, T[$$

`\mathopen` e `\mathclose` non estendono i delimitatori

Due punti :

Bisogna distinguere i due punti ottenuti con `:` (operazione binaria) da quelli ottenuti con `\colon` (simbolo di interpunzione)

Operazione binaria (divisione)

```
\[  
  a:b  
\]
```

$$a : b$$

Simbolo di interpunzione

```
\[  
  f\colon \mathbb{R} \to \mathbb{R}  
\]
```

$$f : \mathbb{R} \rightarrow \mathbb{R}$$

Il simbolo `:` può essere utilizzato anche come abbreviazione di “tale che”

Punti ellittici

I punti ellittici (di sospensione) ... si ottengono nel testo indifferentemente con `\ldots` oppure `\dots`. In ambiente matematico i comandi `\ldots` (...) e `\cdots` (...) producono puntini rispettivamente bassi e centrati, mentre il comando `\dots` li adatta alla situazione

Per scrivere le liste si usano i puntini bassi

```
\[
  1, 2, \ldots, n
\]
```

mentre per scrivere una somma quelli centrati

```
\[
  1 + 2 + \cdots + n.
\]
```

Col comando `\verb|\dots|` ciò avviene automaticamente

```
\[
  1, 2, \dots, n \quad 1 + 2 + \dots + n.
\]
```

Per scrivere le liste si usano i puntini bassi

$$1, 2, \dots, n$$

mentre per scrivere una somma quelli centrati

$$1 + 2 + \cdots + n.$$

Col comando `\dots` ciò avviene automaticamente

$$1, 2, \dots, n \quad 1 + 2 + \cdots + n.$$

\LaTeX mette a disposizione molti tipi di frecce, vediamo alcuni

Per definire una funzione si può usare la notazione

```
\[
f\colon \mathbb{R} \to \mathbb{R}
\quad x\mapsto x^2
\]
```

Per definire una funzione si può usare la notazione

$$f: \mathbb{R} \rightarrow \mathbb{R} \quad x \mapsto x^2$$

Caricando **old-arrows** abbiamo frecce con punta più piccola

```
...
\usepackage{old-arrows}
...
Per definire una funzione si può
usare la notazione
\[
f\colon \mathbb{R} \to \mathbb{R}
\quad x\mapsto x^2
\]
```

Per definire una funzione si può usare la notazione

$$f: \mathbb{R} \rightarrow \mathbb{R} \quad x \mapsto x^2$$

Frecce standard

\leftarrow	<code>\leftarrow</code> o <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>
\rightarrow	<code>\rightarrow</code> o <code>\to</code>	\longrightarrow	<code>\longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>
\Uparrow	<code>\Uparrow</code>		
\Downarrow	<code>\Downarrow</code>		

Il pacchetto `old-arrows` ridisegna *tutte* queste frecce con una punta più stretta

Simboli logici

\forall	<code>\forall</code>	\exists	<code>\exists</code>	\nexists	<code>\nexists</code>
\neg	<code>\not</code> o <code>\neg</code>	\vee	<code>\lor</code> o <code>\vee</code>	\wedge	<code>\land</code> o <code>\wedge</code>
\models	<code>\models</code>	\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>
\implies	<code>\implies</code>	\iff	<code>\iff</code>		

```

\[\
\forall x \in \mathbb{R} \quad \text{\code{quad}}
(x \neq 0 \implies
\exists y \in \mathbb{R} : xy=1)
\]

```

$$\forall x \in \mathbb{R} \quad (x \neq 0 \implies \exists y \in \mathbb{R} : xy = 1)$$

Per formule logiche più complesse e per ottimizzare la spaziatura si consiglia l'utilizzo del pacchetto `gn-logic14` (gene-logic)

Simboli di relazione

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> o <code>\le</code>	\geq	<code>\geq</code> o <code>\ge</code>	\neq	<code>\neq</code> o <code>\ne</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\equiv	<code>\equiv</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\parallel	<code>\parallel</code>	\perp	<code>\perp</code>	\approx	<code>\approx</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\cong	<code>\cong</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\Join	<code>\Join</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\asymp	<code>\asymp</code>	\doteq	<code>\doteq</code>	\mid	<code>\mid</code>

Simboli di operatori binari

$+$	<code>+</code>	$-$	<code>-</code>	\star	<code>\star</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\circ	<code>\circ</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\otimes	<code>\otimes</code>
\times	<code>\times</code>	$*$	<code>\ast</code>	\bigcirc	<code>\bigcirc</code>
\sqcap	<code>\sqcap</code>	\sqcup	<code>\sqcup</code>	\bullet	<code>\bullet</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\wr	<code>\wr</code>
\oslash	<code>\oslash</code>	\odot	<code>\odot</code>	\diamond	<code>\diamond</code>
\triangleup	<code>\bigtriangleup</code>	\triangleleft	<code>\lhd</code>	\uplus	<code>\uplus</code>
\triangledown	<code>\bigtriangledown</code>	\triangleright	<code>\rhd</code>	\dagger	<code>\dagger</code>
\triangleleft	<code>\triangleleft</code>	\trianglelefteq	<code>\unlhd</code>	\ddagger	<code>\ddagger</code>
\triangleright	<code>\triangleright</code>	\trianglerighteq	<code>\unrhd</code>	\amalg	<code>\amalg</code>

La relazione $a \mid b$ si denota con

```
\[
  a \mid b
\]
```

e $\not\mid$ con $a \not\mid b$, che darebbe una spaziatura sbagliata.

Sia \sim una relazione di equivalenza in un insieme A e sia $a \in A$. Si chiama $[a]$ l'insieme

```
\[
  [a] = \{x \in A \mid x \sim a\}
\]
```

Se prendiamo come relazione di equivalenza l'uguaglianza $=$, allora ogni classe contiene un solo elemento e dunque

```
\[
  \forall a, b \in A \quad a = b \implies [a] \cap [b] = \emptyset
\]
```

Inoltre si ha che

```
\[
  \bigcup_{a \in A} [a] = A
\]
```

Se $ax^2 + bx + c = 0$, con $a \neq 0$, allora

```
\[
  x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\]
```

La relazione a divide b si denota con

$$a \mid b$$

e *non* con $a \nmid b$, che darebbe una spaziatura sbagliata.

Sia \sim una relazione di equivalenza in un insieme A e sia $a \in A$. Si chiama *classe di equivalenza* l'insieme

$$[a] = \{x \in A \mid x \sim a\}$$

Se prendiamo come relazione di equivalenza l'uguaglianza $=$, allora ogni classe contiene un solo elemento e dunque

$$\forall a, b \in A \quad a \neq b \implies [a] \cap [b] = \emptyset$$

Inoltre si ha che

$$\bigcup_{a \in A} [a] = A$$

Se $ax^2 + bx + c = 0$, con $a \neq 0$, allora

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Alcune relazioni binarie \mathcal{AMS}

Relazioni \mathcal{AMS}

\backslash	<code>\leqslant</code>	\gg	<code>\geqslant</code>	\equiv	<code>\risingdotseq</code>
\backslash	<code>\eqslantless</code>	\backslash	<code>\eqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\forall	<code>\leqq</code>	\forall	<code>\geqq</code>	\equiv	<code>\eqcirc</code>
\simeq	<code>\lesssim</code>	\simeq	<code>\gtrsim</code>	\triangleq	<code>\triangleq</code>
\approx	<code>\lessapprox</code>	\approx	<code>\gtrapprox</code>	\bumpeq	<code>\bumpeq</code>
\lessgtr	<code>\lessgtr</code>	\lessgtr	<code>\gtrless</code>	\Bumpeq	<code>\Bumpeq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\lesseqgtr	<code>\gtreqless</code>	\thicksim	<code>\thicksim</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\lesseqqgtr	<code>\gtreqqless</code>	\thickapprox	<code>\thickapprox</code>
\prec	<code>\precapprox</code>	\succ	<code>\succapprox</code>	\approx	<code>\approxeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>	\backsimeq	<code>\backsimeq</code>
\prec	<code>\precsim</code>	\succ	<code>\succsim</code>	\backsimeq	<code>\backsimeq</code>
\blacktriangleleft	<code>\blacktriangleleft</code>	\therefore	<code>\therefore</code>	\varpropto	<code>\varpropto</code>
\blacktriangleright	<code>\blacktriangleright</code>	\because	<code>\because</code>	\between	<code>\between</code>

Alcune relazioni binarie \mathcal{AMS} negate

Relazioni \mathcal{AMS} negate

\nless	<code>\nless</code>	\ngtr	<code>\ngtr</code>	\subsetneq	<code>\subsetneq</code>
\lneq	<code>\lneq</code>	\gneq	<code>\gneq</code>	\supsetneq	<code>\supsetneq</code>
\nleq	<code>\nleq</code>	\ngeqslant	<code>\ngeqslant</code>	\nsubseteq	<code>\nsubseteq</code>
\nleqslant	<code>\nleqslant</code>	\ngeqslant	<code>\ngeqslant</code>	\nsupseteq	<code>\nsupseteq</code>
\lneqq	<code>\lneqq</code>	\gneqq	<code>\gneqq</code>	\subsetneqq	<code>\subsetneqq</code>
\nleqq	<code>\nleqq</code>	\ngeqq	<code>\ngeqq</code>	\supsetneqq	<code>\supsetneqq</code>
\lnsim	<code>\lnsim</code>	\gnsim	<code>\gnsim</code>	\nmid	<code>\nmid</code>
\lnapprox	<code>\lnapprox</code>	\gnapprox	<code>\gnapprox</code>	\nparallel	<code>\nparallel</code>
\nprec	<code>\nprec</code>	\nsucc	<code>\nsucc</code>	\nsim	<code>\nsim</code>
\nvDash	<code>\nvDash</code>	\nvDash	<code>\nvDash</code>	\ncong	<code>\ncong</code>

Se per un simbolo non esiste il corrispondente negato, questo si può ottenere anteponendo `\not`

Per negare i simboli `=$`, `=$\in$`, `=$\mid$` esistono specifici comandi e si può scrivere `=\ne`, `=\notin`, `~$ \nmid$`.

Non esistono però i simboli negati, ad esempio, di `=\ni` e `=\subset`.

Essi si possono ottenere anteponendo `\verb|\not|`

```
\[
\not\ni \quad \not\subset
\]
```

Per negare i simboli $=$, \in , $|$ esistono specifici comandi e si può scrivere \neq , \notin , \nmid .

Non esistono però i simboli negati, ad esempio, di \ni e \subset . Essi si possono ottenere anteponendo `\not`

\neq \notin

Rimarchiamo che se il simbolo negato esiste è *sempre* meglio usare quello piuttosto che scrivere `\not`

Lettere greche minuscole

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>	ζ	<code>\zeta</code>	η	<code>\eta</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</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>
o	<code>o</code>	π	<code>\pi</code>	ϖ	<code>\varpi</code>	ρ	<code>\rho</code>
ϱ	<code>\varrho</code>	σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>
υ	<code>\upsilon</code>	ϕ	<code>\phi</code>	φ	<code>\varphi</code>	χ	<code>\chi</code>
ψ	<code>\psi</code>	ω	<code>\omega</code>				

Lettere greche maiuscole

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>		

Per alcune lettere (o varianti) usate spesso è conveniente dare delle nuove definizioni, ad esempio nel modo seguente

```
\newcommand{\eps}{\varepsilon}
\renewcommand{\theta}{\vartheta}
\renewcommand{\phi}{\varphi}
\newcommand{\R}{\mathbb{R}}    % per l'insieme dei numeri reali
...
\[
  \forall \eps > 0 \enskip \exists \delta > 0 \colon \forall x \in \R \ ,
  \bigl( \abs{x - x_0} < \delta \implies \abs{f(x) - L} < \eps \bigr)
\]
\[\theta + \phi = \alpha + \beta\]
```

$$\forall \varepsilon > 0 \exists \delta > 0: \forall x \in \mathbb{R} (|x - x_0| < \delta \implies |f(x) - L| < \varepsilon)$$

$$\vartheta + \varphi = \alpha + \beta$$

Accenti matematici

\hat{a}	<code>\hat{a}</code>	\vec{a}	<code>\vec{a}</code>	\tilde{a}	<code>\tilde{a}</code>
\bar{a}	<code>\bar{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>
\acute{a}	<code>\acute{a}</code>	\grave{a}	<code>\grave{a}</code>	\widehat{A}	<code>\widehat{A}</code>
\check{a}	<code>\check{a}</code>	\breve{a}	<code>\breve{a}</code>	\widetilde{A}	<code>\widetilde{A}</code>

Vettori

Per indicare un vettore si possono usare le seguenti notazioni: la freccetta data da `\vec`, il grassetto (tondo) dato da `\mathbf`, il grassetto (corsivo) dato da `\boldsymbol` (norme UNI-ISO)

```
\[  
  \vec{v} \quad \mathbf{v} \quad \boldsymbol{v}  
\]
```


$$\vec{v} \quad \mathbf{v} \quad \boldsymbol{v}$$

Se si vogliono indicare gli estremi del vettore si usa il comando `\overrightarrow`

```
\[  
  \overrightarrow{AB}  
\]
```


$$\overrightarrow{AB}$$

Col pacchetto `old-arrows` il risultato è migliore

```
\usepackage{old-arrows}  
...  
\[  
  \overrightarrow{AB}  
\]
```


$$\overrightarrow{AB}$$

Barre e angoli

I caratteri barrati si ottengono col comando `\bar`

I comandi `\overline` e `\underline` creano barre orizzontali rispettivamente sopra e sotto i loro argomenti

Per denotare gli angoli si usa invece `\widehat`

Il coniugato di $z=x+iy$ è
`\bar{z}=x-iy`.

Sia dato il segmento AB di misura 2

```
\[  
  \overline{AB}=2
```

```
\]
```

La sottolineatura `\underline{C}` è poco usata. Per indicare un angolo di vertice A si può scrivere

```
\[  
  B\widehat{A}C
```

```
\]
```

oppure

```
\[  
  \widehat{BAC}
```

```
\]
```

Il coniugato di $z = x + iy$ è $\bar{z} = x - iy$.

Sia dato il segmento AB di misura 2

$$\overline{AB} = 2$$

La sottolineatura \underline{C} è poco usata. Per indicare un angolo di vertice A si può scrivere

$$B\hat{A}C$$

oppure

$$\widehat{BAC}$$

Scritture “sopra e sotto”

```
\[  
  a\times n =  
  \underbrace{a+\dots+a}_{\text{\$n\$ addendi}}  
\]
```

$$a \times n = \underbrace{a + \dots + a}_{n \text{ addendi}}$$

```
\[  
  a^n =  
  \overbrace{a\cdot a\cdot a\cdot \dots\cdot a}^{\text{\$n\$ fattori}}  
\]
```

$$a^n = \overbrace{a \cdot a \cdot \dots \cdot a}^{n \text{ fattori}}$$

```
\[  
  \lim_{x\to\infty} \frac{x^2+1}{3x} \stackrel{\text{H}}{=} \lim_{x\to\infty} \frac{2x}{3}  
\]
```

$$\lim_{x \rightarrow \infty} \frac{x^2 + 1}{3x} \stackrel{\text{H}}{=} \lim_{x \rightarrow \infty} \frac{2x}{3}$$

```
\[  
  A \stackrel{f}{\longrightarrow} B  
\]
```

$$A \xrightarrow{f} B$$

Simboli misti

\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>
∞	<code>\infty</code>	∂	<code>\partial</code>	∇	<code>\nabla</code>
\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\wp	<code>\wp</code>	\surd	<code>\surd</code>	\prime	<code>\prime</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\star	<code>\bigstar</code>
\lozenge	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\mho	<code>\mho</code>
\sphericalangle	<code>\angle</code>	\sphericalangle	<code>\sphericalangle</code>	\sphericalangle	<code>\measuredangle</code>
$\dot{+}$	<code>\dotplus</code>	\smallsetminus	<code>\smallsetminus</code>	\cdot	<code>\centerdot</code>
\boxplus	<code>\boxplus</code>	\boxminus	<code>\boxminus</code>	\boxtimes	<code>\boxtimes</code>
\boxdot	<code>\boxdot</code>	\circledast	<code>\circledast</code>	\circledash	<code>\circleddash</code>
\flat	<code>\flat</code>	\natural	<code>\natural</code>	\sharp	<code>\sharp</code>
\spadesuit	<code>\spadesuit</code>	\clubsuit	<code>\clubsuit</code>	\diamondsuit	<code>\diamondsuit</code>
\heartsuit	<code>\heartsuit</code>	\spadesuit	<code>\spadesuit</code>	\llcorner	<code>\llcorner</code>
\lrcorner	<code>\lrcorner</code>	\urcorner	<code>\urcorner</code>	\ulcorner	<code>\ulcorner</code>

Il pacchetto cancel

Il pacchetto `cancel` definisce vari comandi per tracciare linee oblique sui loro argomenti

```
\usepackage{cancel}
...
Le frazioni si possono semplificare
nel modo seguente
\[
\frac{\cancel{x+2}}{x\cancel{(x+2)}}=
\frac{1}{x}
\]
oppure anche così
\[
\frac{\bcancel{a}b}{\bcancel{a}}=b
\]
e ricordatevi che
\[
2\le 2
\]
\xcancel{non} è vera!
```

Le frazioni si possono semplificare nel modo seguente

$$\frac{x+2}{x\cancel{(x+2)}} = \frac{1}{x}$$

oppure anche così

$$\frac{\cancel{a}b}{\cancel{a}} = b$$

e ricordatevi che

$$2 \leq 2$$

~~non~~ è vera!

Simboli matematici in grassetto – `\boldsymbol`

Il comando `\mathbf` mette in grassetto tondo le lettere in modalità matematica senza agire sui simboli

```
\[  
  \mathbf{\Delta \quad A+b  
  \quad \alpha\to\delta}  
\]
```

$$\Delta \quad A + b \quad \alpha \rightarrow \delta$$

I pacchetti \mathcal{AMS} definiscono il comando `\boldsymbol` per mettere in grassetto anche i simboli (le lettere in corsivo)

```
\[  
  \boldsymbol{\Delta \quad A+b  
  \quad \alpha\to\delta}  
\]
```

$$\Delta \quad A + b \quad \alpha \rightarrow \delta$$

Simboli matematici in grassetto – `\pmb`

Viene definito inoltre il comando `\pmb` (*poor man's bold – grassetto dei poveri*) che può essere usato se non esiste una versione del carattere in grassetto all'interno del font

Esso funziona stampando più copie dello stesso simbolo leggermente spostate le une dalle altre, creando un grassetto di qualità inferiore

```
\[  
  \pmb{\Delta \quad A+b \quad \quad  
        \alpha\to\delta}  
\]
```

$\Delta \quad A + b \quad \alpha \rightarrow \delta$

Simboli matematici in grassetto – `\pmb`

Viene definito inoltre il comando `\pmb` (*poor man's bold – grassetto dei poveri*) che può essere usato se non esiste una versione del carattere in grassetto all'interno del font

Esso funziona stampando più copie dello stesso simbolo leggermente spostate le une dalle altre, creando un grassetto di qualità inferiore

```
\[  
  \pmb{\Delta \quad A+b \quad \quad  
  \alpha\to\delta}  
\]
```

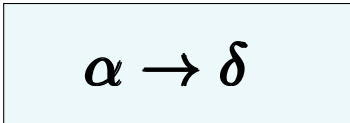


Simboli matematici in grassetto – `\pmb`

Viene definito inoltre il comando `\pmb` (*poor man's bold – grassetto dei poveri*) che può essere usato se non esiste una versione del carattere in grassetto all'interno del font

Esso funziona stampando più copie dello stesso simbolo leggermente spostate le une dalle altre, creando un grassetto di qualità inferiore

```
\[  
  \pmb{\Delta \quad A+b \quad \quad  
        \alpha\to\delta}  
\]
```


$$\alpha \rightarrow \delta$$

Simboli matematici in grassetto – Il pacchetto `bm`

Per risultati di qualità superiore si consiglia l'uso del pacchetto `bm` che definisce il comando `\bm` che oltre a mettere in grassetto i simboli produce correttamente le spaziature (rispetto a `\boldsymbol`)

```
\usepackage{bm}
...
Vediamo in azione il pacchetto
\texttt{bm}
\[
  \bm{\Delta \quad A+b \quad \quad
        \alpha\to\delta}
\]
e facciamo un confronto
\[
  \boldsymbol{T}_1^2 \quad \quad \bm{T}_1^2
\]
per notare le spaziature prodotte.
```

Vediamo in azione il pacchetto `bm`

$$\Delta \quad A + b \quad \alpha \rightarrow \delta$$

e facciamo un confronto

$$T_1^2 \quad T_1^2$$

per notare le spaziature prodotte.

Le matrici si costruiscono con l'ambiente `array` come mostra l'esempio

```
\[
A=\left(\begin{array}{ccc}
0 & -2 & 4 \\
1 & -1 & 0 \\
-5 & 3 & 1
\end{array}\right)
\]
```

$$A = \begin{pmatrix} 0 & -2 & 4 \\ 1 & -1 & 0 \\ -5 & 3 & 1 \end{pmatrix}$$

- ▶ L'argomento `ccc` indica che la matrice ha 3 colonne (la 'c' sta per allineamento verticale centrato)
- ▶ Gli elementi di ogni riga devono essere separati da `&`
- ▶ Il comando `\\` alla fine di ogni riga (tranne eventualmente dell'ultima) fa andare a capo

Altri esempi

Sia data la matrice

```
\[  
  \left[\begin{array}{ccc}  
    x_{11} & x_{12} & x_{13} \\  
    x_{21} & x_{22} & x_{23}\end{array}\right]
```

e anche

```
\[  
  \left[\begin{array}{ccc}  
    2 & -1 & 3 \\  
    0 & 1 & 5 \\  
    \hline  
    1 & 0 & 9\end{array}\right]
```

e calcoliamo il determinante della
sottomatrice

```
\[  
  \left|\begin{array}{cc}  
    2 & -1 \\  
    0 & 1\end{array}\right|=2
```

```
\]
```

Sia data la matrice

$$\left[\begin{array}{cc|c} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{array} \right]$$

e anche

$$\left[\begin{array}{cc|c} 2 & -1 & 3 \\ 0 & 1 & 5 \\ \hline 1 & 0 & 9 \end{array} \right]$$

e calcoliamo il determinante della
sottomatrice

$$\left| \begin{array}{cc} 2 & -1 \\ 0 & 1 \end{array} \right| = 2$$

`amsmath` mette a disposizione vari ambienti per costruire le matrici, con delimitatori predefiniti

Ambiente	Delimitatori
<code>matrix</code>	senza delimitatori
<code>pmatrix</code>	$(.)$
<code>bmatrix</code>	$[.]$
<code>Bmatrix</code>	$\{.\}$
<code>vmatrix</code>	$ \cdot $
<code>Vmatrix</code>	$\ \cdot \ $

l'ambiente `smallmatrix` permette inoltre di inserire matrici in linea col testo

Consideriamo la matrice

```
\[
\begin{bmatrix}
x_{11} & x_{12} & x_{13} \\
x_{21} & x_{22} & x_{23} \\
x_{31} & x_{32} & x_{33}
\end{bmatrix}
\end{bmatrix}
```

e il suo determinante

```
\[
\begin{vmatrix}
x_{11} & x_{12} & x_{13} \\
x_{21} & x_{22} & x_{23} \\
x_{31} & x_{32} & x_{33}
\end{vmatrix}
\end{bmatrix}
```

Consideriamo la matrice

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$$

e il suo determinante

$$\begin{vmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{vmatrix}$$

Come si vede, la spaziatura viene ridotta rispetto ad **array** e non è necessario specificare i parametri relativi alle colonne

Possiamo comporre le matrici anche con punti ellittici

```
\[  
  \begin{bmatrix}  
    x_{11} & x_{12} & \dots \\  
    x_{21} & x_{22} & \dots \\  
    \vdots & \vdots & \ddots  
  \end{bmatrix}
```

oppure

```
\[  
  \begin{bmatrix}  
    a_{11} & \dots & a_{1n} \\  
    a_{21} & \dots & a_{2n} \\  
    \hdotsfor{3} \\  
    a_{n1} & \dots & a_{nn}  
  \end{bmatrix}
```

mentre una matrice in linea

```
appare così  $A = \bigr($   
\begin{smallmatrix}  
  x & y \\  
  z & t  
\end{smallmatrix}\bigr)$.
```

Possiamo comporre le matrici anche con punti ellittici

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

oppure

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

mentre una matrice in linea appare così $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

L'ambiente cases

Per i sistemi e le definizioni per casi è disponibile l'ambiente **cases**

```
\[
\begin{cases}
2x+y=1 \\
3x-2y=-4
\end{cases}
\]
```

$$\begin{cases} 2x + y = 1 \\ 3x - 2y = -4 \end{cases}$$

```
\[
\begin{cases}
\frac{1}{2}x-3y=1 \\
-2x+y=-\frac{4}{5}
\end{cases}
\]
```

$$\begin{cases} \frac{1}{2}x - 3y = 1 \\ -2x + y = -\frac{4}{5} \end{cases}$$

`\\[1em]` manda a capo e aggiunge uno spazio verticale di **1em**

```
\[
f(x)=\begin{cases}
2x+1 & \text{se } x < 0 \\
x^2-1 & \text{se } x \geq 0
\end{cases}
\]
```

$$f(x) = \begin{cases} 2x + 1 & \text{se } x < 0 \\ x^2 - 1 & \text{se } x \geq 0 \end{cases}$$

`amsmath` mette a disposizione molti ambienti utili per costruire le formule in display

Tutti questi ambienti (tranne `split`) hanno una versione asteriscata che non attiva la numerazione

Ambiente	Descrizione
<code>multline</code>	spezza una formula in più righe: la prima è allineata a sinistra, la seconda a destra e le altre centrate
<code>split</code>	spezza una formula in più righe incolonnate; va usato dentro un altro ambiente per formule in display
<code>gather</code>	raggruppa più formule centrando e numerando ciascuna riga
<code>align</code>	incolonna gruppi di formule e numerando ciascuna riga

Le formule con \mathcal{AMS} – multiline

```
\begin{multiline}  
  f=a+b+c \\  
  +d+e+g+h \\  
  +r+s+t  
\end{multiline}
```

$$\begin{aligned} f &= a + b + c \\ &+ d + e + g + h \\ &+ r + s + t \quad (1.1) \end{aligned}$$

La versione asteriscata `multiline*` non numera la formula

```
\begin{multiline*}  
  f=a+b+c \\  
  +d+e+g+h \\  
  +r+s+t  
\end{multiline*}
```

$$\begin{aligned} f &= a + b + c \\ &+ d + e + g + h \\ &+ r + s + t \end{aligned}$$

Le formule con AMS – split

L'ambiente `split` deve essere inserito all'interno di un altro ambiente di display matematico, il quale si occuperà della numerazione (a meno che sia asteriscato oppure sia `\[\]`)

```
\begin{equation}
\begin{split}
a &= b+c-d \\
&= e-f \\
&= g
\end{split}
\end{equation}
```

$$\begin{aligned} a &= b + c - d \\ &= e - f \\ &= g \end{aligned} \tag{1.1}$$

```
\[
\begin{split}
a &= b+c-d \\
&= e-f \\
&= g
\end{split}
\]
```

$$\begin{aligned} a &= b + c - d \\ &= e - f \\ &= g \end{aligned}$$

Le formule con \mathcal{AMS} – gather

```
\begin{gather}  
  a = b + c \\  
  x = y + z + t  
\end{gather}
```

$$a = b + c \quad (1.1)$$
$$x = y + z + t \quad (1.2)$$

Ogni formula viene numerata. Se per qualcuna di esse non si vuole la numerazione basta scrivere alla fine della formula in questione il comando `\notag`

```
\begin{gather}  
  a = b + c \notag \\  
  x = y + z + t  
\end{gather}
```

$$a = b + c$$
$$x = y + z + t \quad (1.1)$$

La versione asteriscata `gather*` non numera alcuna formula

Le formule con \mathcal{AMS} – align

```
\begin{align}
  a &= b + c \\
  x + y &= z + t \ \backslash\text{notag} \\
  w &= v + p + s
\end{align}
```

$$a = b + c \quad (1.1)$$

$$x + y = z + t$$

$$w = v + p + s \quad (1.2)$$

Per evitare la numerazione si usa il comando `\notag`

Si possono incolonnare più gruppi di formule:

```
\begin{align}
  x &= y & X &= Y \\
  x' &= y' & X' &= Y' \\
  x+x' &= y+y' & X+X' &= Y+Y'
\end{align}
```

$$x = y \quad X = Y \quad (1.1)$$

$$x' = y' \quad X' = Y' \quad (1.2)$$

$$x + x' = y + y' \quad X + X' = Y + Y' \quad (1.3)$$

La versione asteriscata `align*` non numera alcuna formula

Le formule con \mathcal{AMS} – align

```
\begin{align}
x &= y_1 - y_2 + y_3 - y_5 + y_8 - \dots && \text{\textit{per} \eqref{eq:C}} \\
&= y' \text{\textcircled{y}^*} && \text{\textit{per} \eqref{eq:D}} \\
&= y(0) y' && \text{\textit{per} l'Assioma 1}
\end{align}
```

$$\begin{aligned} x &= y_1 - y_2 + y_3 - y_5 + y_8 - \dots && \text{per (3.21)} && (4.15) \\ &= y' \circledast y^* && \text{per (4.1)} && (4.16) \\ &= y(0)y' && \text{per l'Assioma 1} && (4.17) \end{aligned}$$

Le formule con \mathcal{AMS} – Varianti gathered e aligned

Le varianti **gathered** e **aligned** consentono di ottenere strutture adatte per posizionare parentesi con eventuale testo esplicativo

```
\usepackage{esint}
...
\[
\left.\begin{gathered}
\oint \vec{E} \cdot d\vec{s} = -\frac{d\Phi_B}{dt} \quad \ll[1em] \\
\oiint_S \vec{B} \cdot d\vec{S} = 0
\end{gathered}\right\} \text{equazioni di Maxwell}
\]
```

$$\left. \begin{aligned} \oint \vec{E} \cdot d\vec{s} &= -\frac{d\Phi_B}{dt} \\ \oiint_S \vec{B} \cdot d\vec{S} &= 0 \end{aligned} \right\} \text{equazioni di Maxwell}$$

Le convenzioni tipografiche della scrittura matematica sono consolidate e stabilite dalla tradizione

Nei documenti tecnico-scientifici redatti da fisici e ingegneri si devono seguire di legge le indicazioni fornite dalle norme UNI (*Ente Nazionale Italiano di Unificazione*) e ISO (*International Standard Organization*), in particolare la norma ISO 80000

Tuttavia, se il documento non deve avere valore legale, i matematici “puri” si discostano dalle norme ISO poiché

“La tradizione tipografica per la matematica pura è in netto contrasto con le norme ISO non tanto per ragioni ‘sentimentali’, quanto per motivi attinenti alla matematica stessa”

Enrico Gregorio

Le norme UNI-ISO stabiliscono che:

- ▶ le variabili (come x, y, \dots), i parametri (come a, b, \dots), gli indici variabili (come i in $\sum_i x_i$) e i simboli per le funzioni (come f, g, \dots) vanno scritti in corsivo;
- ▶ i nomi espliciti delle funzioni e degli operatori (come $\sin, \cos, \log, \operatorname{div}, \dots$) vanno scritti in tondo;
- ▶ le grandezze e le costanti fisiche vanno scritte in corsivo; i numeri e le rispettive unità di misura vanno scritte in tondo:

$$i = 2,0 \text{ A} \quad g = 9,8 \text{ m/s}^2 \quad c = 299\,792\,458 \text{ m/s}^2$$

- ▶ le costanti matematiche, come e, i, π , e la d del differenziale, come in df , vanno scritte in tondo (**Attenzione: questo è contrario alla tradizione matematica!**)

Differenze fra UNI-ISO e tradizione matematica

Elementi	UNI-ISO	tradizione matematica
costanti matematiche	$e, i, \pi, \varphi, \gamma$	$e, i, \pi, \varphi, \gamma$
variabili matematiche	x, y, Δ, Γ	x, y, Δ, Γ
differenziali	$\frac{dy}{dx}, \frac{df}{dx}, \int_a^b f(x) dx$	$\frac{dy}{dx}, \frac{df}{dx}, \int_a^b f(x) dx$

In ambiente matematico le lettere in tondo si ottengono col comando `\mathrm`; i caratteri greci in tondo si possono ottenere col pacchetto `pm-isomath` (basta aggiungere `\ISO` al nome della lettera)

```
...  
\usepackage{pm-isomath} % per il greco  
                        % tondo
```

```
...  
Le costanti matematiche, secondo le  
norme UNI-ISO, si scrivono in tondo
```

```
\[  
  \mathrm{e} \quad \mathrm{i} \quad \quad \quad  
  \ISOpi \quad \ISOphi
```

```
\]  
mentre secondo la tradizione  
matematica si scrivono in corsivo
```

```
\[  
  e \quad i \quad \quad \quad \pi \quad \quad \varphi  
\]
```

Le costanti matematiche, secondo le norme UNI-ISO, si scrivono in tondo

$e \quad i \quad \pi \quad \varphi$

mentre secondo la tradizione matematica si scrivono in corsivo

$e \quad i \quad \pi \quad \varphi$

Il pacchetto `pm-isomath` imposta di default, in ambiente matematico, le lettere greche maiuscole in corsivo; se non viene caricato tale pacchetto, esse si possono ottenere antepponendo al nome della lettera il prefisso `\var`

Le variabili, secondo le norme UNI-ISO, si scrivono in corsivo

```
\[  
x \quad y \quad \quad \varDelta  
\quad \varGamma \quad \quad \varTheta  
\]
```

proprio come nella tradizione matematica, tranne che per le lettere greche maiuscole, che si scrivono in tondo

```
\[  
x \quad y \quad \quad \Delta  
\quad \Gamma \quad \quad \Theta  
\]
```

Le variabili, secondo le norme UNI-ISO, si scrivono in corsivo

$$x \quad y \quad \Delta \quad \Gamma \quad \Theta$$

proprio come nella tradizione matematica, tranne che per le lettere greche maiuscole, che si scrivono in tondo

$$x \quad y \quad \Delta \quad \Gamma \quad \Theta$$

I differenziali secondo le norme ISO possono essere ottenuti in modo semplice definendo il seguente nuovo comando `\diff`

```
\newcommand*{\diff}{\mathop{}}\!\mathrm{d}} % differenziale  
% in tondo
```

```
...  
\newcommand*{\diff}{\mathop{}}\!\mathrm{d}}  
...  
\begin{document}  
Le derivate e gli integrali si scrivono,  
secondo le norme ISO, con la 'd' tonda  
\[  
  \frac{\diff y}{\diff x} \quad \int_a^b f(x)\diff x  
\]  
mentre la tradizione matematica vuole il  
corsivo  
\[  
  \frac{dy}{dx} \quad \int_a^b f(x)\,dx  
\]
```

Le derivate e gli integrali si scrivono, secondo le norme ISO, con la 'd' tonda

$$\frac{dy}{dx} \quad \int_a^b f(x) dx$$

mentre la tradizione matematica vuole il corsivo

$$\frac{dy}{dx} \quad \int_a^b f(x) dx$$

Scrivere i numeri

I numeri devono sempre essere scritti in tondo, anche in un testo in corsivo. Quindi è sempre consigliabile scrivere i numeri in ambiente matematico

Sembrava che la distanza fosse di `$5\unit{km}$`, ma era `\emph{solo di $3\unit{km}$}` in linea d'aria!

Sembrava che la distanza fosse di 5 km, ma era *solo di 3 km in linea d'aria!*

È consuetudine, quando si scrivono numeri con molte cifre, raggruppare le cifre a tre a tre separandole con uno spazio fine `\`, tranne nel caso di 4 cifre

Si scrive `$1\,254\,000$` oppure `$25\,782$`, ma `\emph{non} $1\,874$`, bensì `1874`.

Nei testi i numeri inferiori a venti vanno in lettere.

Si scrive 1 254 000 oppure 25 782, ma *non* 1 874, bensì 1874.

Nei testi i numeri inferiori a venti vanno in lettere.

Scrivere i numeri – La virgola intelligente

\LaTeX usa di default il punto come separatore decimale; la virgola è un segno di interpunzione anche in ambiente matematico e produce uno spazio dopo di sé. Si può ovviare attivando la “virgola intelligente” fornita da `babel` con l’opzione `italian`, inserendo la dichiarazione `\IntelligentComma` che funziona così:

- ▶ se dopo la virgola inseriamo un carattere numerico, la virgola viene interpretata come separatore decimale
- ▶ se dopo la virgola inseriamo un carattere non numerico, la virgola viene interpretata come simbolo di interpunzione. In particolare, se stiamo compilando un elenco di numeri, bisogna ricordarsi di inserire uno spazio dopo la virgola

Per disattivare la virgola intelligente basta dare il comando `\NoIntelligentComma`

Se scriviamo una coppia ordinata o una lista

```
\[
(1,2) \quad\text{o}\quad 1,2,\ldots
\]
```

`\LaTeX` interpreta la virgola come segno di interpunzione. Sebbene si possa usare il “trucco”

```
\[
12{,}7
\]
```

se attiviamo la virgola intelligente, `\IntelligentComma` e scriviamo

```
\[
12,7 \quad\text{o}\quad (a,b)
\]
```

abbiamo la virgola decimale e il simbolo di interpunzione, rispettivamente. Però se vogliamo la virgola di interpunzione tra due caratteri numerici dobbiamo mettere uno spazio

```
\[
(1, 2) \quad\text{e } \emph{non}\quad (1,2)
\]
```

Infine, se disattiviamo la virgola intelligente

```
\NoIntelligentComma
ritorniamo ad avere
```

```
\[
12,7
\]
```

Se scriviamo una coppia ordinata o una lista

$(1,2)$ o $1,2,\dots$

`LATEX` interpreta la virgola come segno di interpunzione. Sebbene si possa usare il “trucco”

12,7

se attiviamo la virgola intelligente, e scriviamo

12,7 o (a,b)

abbiamo la virgola decimale e il simbolo di interpunzione, rispettivamente. Però se vogliamo la virgola di interpunzione tra due caratteri numerici dobbiamo mettere uno spazio

$(1,2)$ e *non* $(1,2)$

Infine, se disattiviamo la virgola intelligente ritorniamo ad avere

12,7

Gradi sessagesimali e gradi Celsius

Il carattere ° non è direttamente disponibile da tastiera

Ci sono comunque diverse possibilità per ottenere il simbolo dei gradi: la combinazione matematica `^\circ`, il pacchetto `siunitx` e il pacchetto `gensymb` (eventualmente con `newunicodechar`)

```
\setISOcompliance
...
\[
  60^\circ \quad 10^\circ\text{C}
\]
```

60° 10°C

```
\usepackage{siunitx}
...
\[
  \SI{60}{\degree} \quad \SI{10}{\celsius}
\]
```

60° 10°C

Da notare la spaziatura differente nei gradi sessagesimali e Celsius

Il pacchetto `gensymb` rende disponibili, come `siunitx`, i comandi `\degree` e `\celsius`

```
\usepackage{gensymb}

\setISOcompliance
...
\[
  60\degree \quad \quad 10\unit{\celsius}
\]
```

60° 10 °C

La soluzione migliore è usare le funzionalità del pacchetto `newunicodechar` che permette di inserire ° direttamente da tastiera

```
\usepackage{gensymb}
\usepackage{newunicodechar}

\newunicodechar{°}{\degree}

\setISOcompliance
...
\[
  60° \quad \quad 10\unit{°C}
\]
```

60° 10 °C

Apici e pedici vanno scritti in corsivo se rappresentano variabili, ma vanno scritti in tondo se sono apposizioni della grandezza fisica

L'opzione `italian` di `babel` mette a disposizione per questo scopo i comandi `\ap` e `\ped`

```
\usepackage[italian]{babel}
...
Indichiamo il volume alla
temperatura  $T$  (variabile) col
simbolo
\[\
  V_T
\]
mentre il volume iniziale con
\[\
  V\ped{i}
\]
L'energia cinetica massima
sarà indicata con
\[\
  K\ap{max}
\]
e \emph{non} con  $K^{\max}$ .
```

Indichiamo il volume alla temperatura T (variabile) col simbolo

$$V_T$$

mentre il volume iniziale con

$$V_i$$

L'energia cinetica massima sarà indicata con

$$K^{\max}$$

e *non* con K^{max} .

Gli enunciati dei teoremi, proposizioni, lemmi, corollari e definizioni devono avere un formato e una numerazione particolari

Il pacchetto `amsthm` mette a disposizione diversi strumenti per gestire gli enunciati in tutti i loro aspetti

Nel preambolo del documento dovremo definire ogni tipo di enunciato (ad es. teorema e corollario) come segue

```
\newtheorem{teo}{Teorema}  
\newtheorem{cor}{Corollario}
```

Avremo così a disposizione gli ambienti `teo` e `cor` che consentono di scrivere teoremi e corollari nei formati desiderati con una propria numerazione indipendente

```

...
\newtheorem{teo}{Teorema}
\newtheorem{cor}{Corollario}
...
\begin{document}
Enunciamo ora un importante risultato.

\begin{teo}
Sopra la panca la capra campà.
\end{teo}

I prossimi enunciati sono corollari del
precedente.

\begin{cor}
Sotto la panca la capra crepà.
\end{cor}

\begin{cor}
Di fianco non si sa.
\end{cor}

```

Enunciamo ora un importante risultato.

Teorema 1. *Sopra la panca la capra campà.*

I prossimi enunciati sono corollari del precedente.

Corollario 1. *Sotto la panca la capra crepà.*

Corollario 2. *Di fianco non si sa.*

Come si vede teoremi e corollari hanno un formato speciale (testo in corsivo e spaziatura prima e dopo) e seguono una numerazione indipendente

Se si vuole che i corollari seguano la stessa numerazione dei teoremi bisogna specificare l'argomento opzionale `teo` (che identifica la numerazione dei teoremi) nella definizione di `cor`

```
...
\newtheorem{teo}{Teorema}
\newtheorem{cor}[teo]{Corollario}
...
\begin{document}
Enunciamo ora un importante risultato.

\begin{teo}
Sopra la panca la capra campa.
\end{teo}

I prossimi enunciati sono corollari del
precedente.

\begin{cor}
Sotto la panca la capra crepa.
\end{cor}

\begin{cor}
Di fianco non si sa.
\end{cor}
```

Enunciamo ora un importante risultato.

Teorema 1. *Sopra la panca la capra campa.*

I prossimi enunciati sono corollari del precedente.

Corollario 2. *Sotto la panca la capra crepa.*

Corollario 3. *Di fianco non si sa.*

Se in più si vuole che la numerazione dei teoremi segua ad esempio quella dei paragrafi (in modo che il primo teorema del paragrafo 4 abbia numero 4.1) si deve specificare l'argomento opzionale `section` nella definizione di `teo`

```
...
\newtheorem{teo}{Teorema}[section]
\newtheorem{cor}[teo]{Corollario}
...
\section{Le capre e le panche}
```

Enunciamo ora un importante risultato.

```
\begin{teo}
Sopra la panca la capra campa.
\end{teo}
```

I prossimi enunciati sono corollari del precedente.

```
\begin{cor}
Sotto la panca la capra crepa.
\end{cor}
```

```
\begin{cor}
Di fianco non si sa.
\end{cor}
```

4 Le capre e le panche

Enunciamo ora un importante risultato.

Teorema 4.1. *Sopra la panca la capra campa.*

I prossimi enunciati sono corollari del precedente.

Corollario 4.2. *Sotto la panca la capra crepa.*

Corollario 4.3. *Di fianco non si sa.*

Le definizioni di enunciati possono essere date con le seguenti dichiarazioni, mutuamente esclusive

```
\newtheorem{⟨id-enunciato⟩}{⟨titolo⟩}[⟨sezione⟩]
```

```
\newtheorem{⟨id-enunciato⟩}[⟨numerato come⟩]{⟨titolo⟩}
```

- ▶ *⟨id-enunciato⟩* è la parola chiave che identifica il tipo di enunciato (nome dell'ambiente)
- ▶ *⟨titolo⟩* è il titolo con cui compare l'enunciato nel documento
- ▶ *⟨sezione⟩* è il livello di sezionamento a cui è collegata la numerazione (generalmente **chapter** o **section**)
- ▶ *⟨numerato come⟩* indica il tipo di enunciato la cui numerazione deve essere seguita

L'enunciato si compone inserendolo nell'ambiente $\langle id-enunciato \rangle$, eventualmente con una specificazione

```
\begin{\langle id-enunciato \rangle} [\langle eventuale specificazione \rangle]
...
\end{\langle id-enunciato \rangle}
```

```
...
\newtheorem{teo}{Teorema}[section]
...
\begin{teo}[di Weierstrass]
Ogni funzione reale definita in un
intervallo chiuso e limitato ha
massimo e minimo assoluti.
\end{teo}
```

Teorema 3.2 (di Weierstrass). *Ogni funzione reale definita in un intervallo chiuso e limitato ha massimo e minimo assoluti.*

La versione asteriscata `\newtheorem*` non attiva la numerazione

```
...
\newtheorem*{LCS}{Legge di
    conservazione dello sporco}
\newtheorem*{cor}{Corollario}
...
\begin{LCS}
Per pulire una cosa bisogna sporcarne
un'altra.
\end{LCS}

\begin{cor}
Ma si può sporcare tutto senza pulire
niente.
\end{cor}
```

Legge di conservazione dello sporco. *Per pulire una cosa bisogna sporcarne un'altra.*

Corollario. *Ma si può sporcare tutto senza pulire niente.*

Stili per gli enunciati

Ci sono tre stili predefiniti per gli enunciati, ciascuno con caratteristiche diverse e adatti per situazioni diverse, selezionabili col comando `\theoremstyle`

Stile	Descrizione
<code>plain</code>	per teoremi, lemmi, corollari, proposizioni, congetture, criteri, leggi; è lo stile di default e mette il contenuto in corsivo
<code>definition</code>	per definizioni ed esempi; mette il contenuto in tondo
<code>remark</code>	per commenti, note, osservazioni; mette il contenuto in tondo

```

\theoremstyle{plain}           % default (si può omettere, ma questo
\newtheorem{teo}{Teorema}[section] % gruppo va scritto per primo)
\newtheorem{prop}[teo]{Proposizione}

\theoremstyle{definition}
\newtheorem{definiz}[teo]{Definizione}

\theoremstyle{remark}
\newtheorem{oss}{Osservazione}[section]
...
\section{Radici e dintorni}

\begin{prop}\label{pr:2sol}
Dato un numero reale positivo  $a$ , l'equazione
  \begin{equation}\label{eq:2sol}
x^2-a=0
  \end{equation}
ha due soluzioni reali di segno opposto.
\end{prop}

\begin{definiz}
Nelle condizioni della proposizione~\ref{pr:2sol}, chiamiamo la
soluzione positiva dell'equazione radice quadrata di  $a$ .
\end{definiz}

\begin{oss}
Chiariamo che la soluzione negativa dell'equazione~\eqref{eq:2sol} non
è la radice quadrata di  $a$ . Per denotarla possiamo scrivere  $-\sqrt{a}$ .
\end{oss}

```

1 Radici e dintorni

Proposizione 1.1. *Dato un numero reale positivo a , l'equazione*

$$x^2 - a = 0 \tag{1}$$

ha due soluzioni reali di segno opposto.

Definizione 1.2. Nelle condizioni della proposizione 1.1, chiamiamo la soluzione positiva dell'equazione *radice quadrata di a* .

Osservazione 1.1. Chiariamo che la soluzione negativa dell'equazione (1) non è la radice quadrata di a . Per denotarla possiamo scrivere $-\sqrt{a}$.

Il comando `\swapnumbers` posto all'inizio delle dichiarazioni `\newtheorem` permette di porre la numerazione degli enunciati prima del titolo

```
\swapnumbers
\newtheorem{teo}{Teorema}[section] % stile plain
                                   % automatico

\theoremstyle{definition}
\newtheorem{definiz}[teo]{Definizione}
...
\section{Il piano complesso}

\begin{definiz}
Chiamiamo piano complesso esteso l'insieme
\[\overline{\mathbb{C}}=\mathbb{C}\cup\{\infty\}\]
\end{definiz}

\begin{teo}
Se  $\{a_n\}$  e  $\{b_n\}$  sono due successioni
infinitesime, anche la loro somma è
infinitesima.
\end{teo}
```

1 Il piano complesso

1.1 Definizione. Chiamiamo piano complesso esteso l'insieme

$$\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$$

1.2 Teorema. Se $\{a_n\}$ e $\{b_n\}$ sono due successioni infinitesime, anche la loro somma è infinitesima.

L'ambiente `proof` produce l'intestazione “Dimostrazione” e la chiusura con il simbolo \square (che sta per QED), nonché le spaziature appropriate

```
\begin{teo}
Dato un numero  $x \in \mathbb{R}$ , si ha
che  $x+1 > x$ .
\end{teo}
```

```
\begin{proof}
Per le proprietà dell'ordinamento dei
numeri reali, a partire dalla
relazione  $1 > 0$  basta aggiungere  $x$ 
a entrambi i membri.
\end{proof}
```

Teorema 1. *Dato un numero $x \in \mathbb{R}$, si ha che $x + 1 > x$.*

Dimostrazione. Per le proprietà dell'ordinamento dei numeri reali, a partire dalla relazione $1 > 0$ basta aggiungere x a entrambi i membri. \square

È possibile personalizzare sia l'intestazione che il simbolo di chiusura (prodotto dal comando `\qedsymbol`)

```
\newtheorem*{TeoGin}{Teorema di Ginsberg}
...
\begin{TeoGin}[TG]
Non puoi vincere, non puoi pareggiare,
non puoi nemmeno abbandonare.
\end{TeoGin}

\renewcommand{\qedsymbol}{\blacksquare}

\begin{proof}[Dimostrazione di TG]
Basta provare a vivere un po' di vita
per convincersi della validità
dell'asserto.
\end{proof}
```

Teorema di Ginberg (TG). *Non puoi vincere, non puoi pareggiare, non puoi nemmeno abbandonare.*

Dimostrazione di TG. Basta provare a vivere un po' di vita per convincersi della validità dell'asserto. ■

Se una dimostrazione termina con una formula in display è possibile usare il comando `\qedhere` per posizionare il simbolo di chiusura allo stesso livello della formula

```
\newtheorem{prop}{Proposizione}
...
\begin{prop}
Dati due numeri reali  $x$  e  $y$ , se  $xy=0$ 
e  $x \neq 0$ , allora  $y=0$ .
\end{prop}

\begin{proof}
Se  $x \neq 0$ , allora esiste  $x^{-1}$  tale
che  $x^{-1}x=1$ . Vale la seguente catena di
uguaglianze
\[
y = x^{-1}xy = x^{-1} \cdot 0 = 0 \quad \text{\qedhere}
\]
\end{proof}
```

Proposizione 1. *Dati due numeri reali x e y , se $xy = 0$ e $x \neq 0$, allora $y = 0$.*

Dimostrazione. Se $x \neq 0$, allora esiste x^{-1} tale che $x^{-1}x = 1$. Vale la seguente catena di uguaglianze

$$y = x^{-1}xy = x^{-1} \cdot 0 = 0 \quad \square$$

Ancora sui font Blackboard Bold

Sappiamo che il comando `\mathbb` produce i font \mathcal{AMS} Blackboard Bold

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Per avere uno stile diverso per questi font è possibile scrivere *nel preambolo* il seguente codice

```
\DeclareFontEncoding{LS1}{}{}  
\DeclareFontSubstitution{LS1}{stix}{m}{n}  
\DeclareMathAlphabet{\mathbb}{LS1}{stixbb}{m}{n}
```

che preleva i Blackboard Bold dai font STIX associandoli sempre al comando `\mathbb`

ABCDEFGHIJKLMNOPQRSTUVWXYZ

e aggiungendo inoltre i caratteri

01234567890abcdefghijklmnopqrstuvwxyz

Simili ai Blackboard Bold STIX ci sono i font Universal Mathematical Pi #6, che però sono a pagamento e dunque non forniti insieme alle distribuzioni T_EX

Il pacchetto `mathpi` con l'opzione `mathbb` permette comunque il loro utilizzo in L^AT_EX (dopo averli installati manualmente)

```
\usepackage[mathbb]{mathpi}
...
\[
  \mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
\]
```

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Riuscite a notare le differenze?

NZQRC

stix

NZQRC

mathpi

Grazie dell'attenzione!



Happy T_EXing!