

PCTO L^AT_EX

Grazia Messineo — Salvatore Vassallo

Maggio - Giugno 2026

Indice I

T_EX e L_AT_EX

Cosa sono

Come funziona L_AT_EX?

L_AT_EX online

Il documento

La compilazione

Preambolo

Il testo

I caratteri

Comandi

Scrivere

Ambienti

Le note

Figure

Le scatole

Le lunghezze in L_AT_EX

La gestione della pagina

Indice II

- Il codice

- Le macro di $\text{\LaTeX}$

- $\text{\LaTeX}$ e la matematica

- $\text{\LaTeX}$ e le altre scienze

- Il sezionamento

- Gli oggetti flottanti

- I riferimenti incrociati

- La bibliografia

- Gli indici

- Le presentazioni: il pacchetto `beamer`

- Il preambolo

- Il corpo del documento

- La grafica

- PSTricks

- TikZ e PGF

- Le forme geometriche di base

- Grafici di funzioni

Indice III

Nodi e testo

Per saperne di più

L^AT_EX è un linguaggio di “markup” usato per la preparazione di testi, basato sul programma di composizione tipografica T_EX.

T_EX è stato scritto dal matematico ed informatico Donald Knuth



T_EX è oggi il solo programma che produce testo matematico, scritto e impaginato esattamente come i matematici lo vogliono.

Il numero di versione di T_EX tende a π e attualmente (3 maggio 2024) è 3.141592653.



L^AT_EX è stato scritto dal matematico ed informatico Leslie Lamport e contiene i comandi per un utilizzo comodo di T_EX insieme a delle impostazioni predefinite per l'impaginazione del testo.

Osservazione

La pronuncia del nome non è “tex”, ma “tek” (o meglio “tech” con il “ch” come in “Bach”) , poiché l'origine del nome T_EX deriva dalla radice della parola greca τέχνη (téchnē), che significa “arte, tecnica, tecnologia”. La terminazione del nome è la lettera greca chi maiuscola (X) la cui grafia si confonde con quella della latina “ics” (X).

Cosa serve

- ▶ Per usare $\text{\LaTeX}$ è necessario un computer, qualsiasi computer con qualsiasi sistema operativo;
- ▶ Una distribuzione $\text{\TeX}$ ($\text{\TeX}$ Live);
- ▶ Un editor di testo.

In realtà per iniziare basta un computer connesso ad internet.

La filosofia

$\text{\LaTeX}$ prevede che un documento sia strutturato in *unità logiche*: molti comandi di $\text{\LaTeX}$ descrivono la struttura logica del documento, cioè il documento stesso, i capitoli, le sezioni, . . .

$\text{\LaTeX}$ è concepito in modo che l'utente si concentri sul contenuto anziché sulla forma

Lavorare con $\text{\LaTeX}$

$\text{\LaTeX}$ produce il suo output da un file di testo (file `tex`) mediante una *compilazione* con un programma (`pdflatex`, `xelatex`, `lualatex`): l'output è (di solito) un file pdf che può essere stampato o visualizzato a schermo.

Quando si compila un documento, $\text{\LaTeX}$ genera sempre oltre al file di output altri due file:

- ▶ un file `.log`, che contiene una copia dei messaggi che appaiono durante la compilazione (utile, per esempio, quando vengono rilevati errori di compilazione);
- ▶ un file `.aux`, che contiene gli eventuali riferimenti incrociati (bibliografia, note, ecc.).

Lavorare con $\text{\LaTeX}$

A seconda della complessità del documento, possono essere creati anche altri file:

- ▶ il file `.toc`, che contiene l'indice;
- ▶ il file `.lot`, che contiene la lista delle tabelle;
- ▶ il file `.lof`, che contiene la lista delle figure;
- ▶ il file `.idx`, che contiene l'indice analitico non formattato.

cocalc

cocalc (<https://cocalc.com/>) non è solo un editor per T_EX, ma una vera e propria piattaforma, utilizzabile in modo collaborativo, con programmi di calcolo, di statistica e la possibilità di utilizzare un ambiente completo Linux

overleaf

overleaf (<https://it.overleaf.com/>) permette l'utilizzo di un editor “user friendly” e di una distribuzione T_EX pressoché completa.

Noi useremo *overleaf*.

Si comincia registrandosi con una mail o con la propria utenza Google.

A questo punto si può creare un progetto cioè il nostro primo documento $\text{\LaTeX}$.

Scegliamo un progetto bianco

Il mio primo documento L^AT_EX

```
\documentclass[a4paper,12pt]{article}
\usepackage[italian]{babel}
\begin{document}
    Questo è il mio primo    ← qui vado a capo
    documento LaTeX.
\end{document}
```

La compilazione - 1

Una volta scritto il documento $\text{T}_{\text{E}}\text{X}$ è necessario *compilarlo* con uno dei programmi di composizione: $\text{pdf}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$, $\text{X}_{\text{Y}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, $\text{Lua}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$, ...

Come si compila? Usando un editor per $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ di solito c'è un'icona o una voce di menu per lanciare la compilazione, su *Overleaf* c'è un pulsante.

Se la compilazione va a buon fine viene prodotto un file pdf

Con molti editor c'è la possibilità di sincronizzare il file tex e il file pdf cioè ci si può spostare sul file pdf al punto corrispondente del file tex e viceversa.

La compilazione - 2

E se la compilazione non va a buon fine?

In questo caso viene prodotto un messaggio d'errore che, se non è visibile direttamente a schermo, può essere letto sul file log che il programma produce.

Gli errori più comuni sono banali errori di battitura: se io scrivo `\textbg` anziché `\textbf`, $\text{\TeX}$ non è in grado di interpretare il comando e si interrompe, spesso un ambiente viene aperto, ma non viene chiuso o, analogamente, viene aperta una parentesi e poi non viene chiusa.

Il consiglio è quello di compilare spesso il documento così che eventuali errori si possano trovare solo in una piccola porzione di testo.

Il mio secondo documento L^AT_EX

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}
\title{Un documento molto grande}
\author{Salvatore Vassallo}\date{}
\begin{document}
\maketitle
Questo è il mio secondo ← qui vado a capo
documento \LaTeX\ ed è molto più lungo e impegnativo del primo.
← qui vado a capo due volte

Sono molto contento di usare \LaTeX\ anche se è difficile, ma
penso che impegnandomi, studiando e svolgendo gli esercizi
riuscirò sicuramente a dominare la materia. \\ ← cosa succede
qui?
Spero che questo corso mi insegni le basi e poi proseguirò
io autonomamente. ← quanti spazi
\end{document}
```

Il mio secondo documento L^AT_EX

Un documento molto lungo

Salvatore Vassallo

Questo è il mio secondo documento L^AT_EX ed è molto più lungo e impegnativo del primo.

Sono molto contento di usare L^AT_EX anche se è difficile, ma penso che impegnandomi, studiando e svolgendo gli esercizi riuscirò sicuramente a dominare la materia.

Spero che questo corso mi insegni le basi e poi proseguirò io autonomamente.

Questo testo è allineato a destra e non è giustificato. Non solo l'aspetto non è bellissimo, ma è più difficile da leggere.

Classe di un documento - 1

Quello che viene scritto di un documento $\text{\LaTeX}$ è quello che appare tra

`\begin{document}` e `\end{document}`

Quello che appare prima è il **preambolo** del documento

La prima istruzione è la dichiarazione di **classe** del documento

Classe

```
\documentclass{article}
```

```
\documentclass{report}
```

```
\documentclass{book}
```

Classe di un documento - 2

Il comando ha un argomento opzionale (tra parentesi quadre)

Le principali opzioni sono

Opzioni della classe

- ▶ **a4paper**: sceglie una **gabbia** adatta al foglio A4 (ci sono **letterpaper**, **a3paper**, ...)
- ▶ **twoside**: prepara la gabbia per la stampa fronte-retro
- ▶ **10pt**, **11pt**, **12pt**: sceglie un carattere **corpo** 10, 11, 12
- ▶ **twocolumn**: per scrivere l'articolo su due colonne
- ▶ **gabbia**= la parte di foglio dove viene scritto il testo. La pagina è formata da **testatina**, **gabbia**, **piè di pagina**
- ▶ **corpo**= misura della grandezza dei caratteri

I pacchetti - 1

Dopo la dichiarazione di classe inizia il preambolo vero e proprio dove vengono specificati i **pacchetti** aggiuntivi che verranno usati

Un pacchetto aggiuntivo può:

- ▶ aggiungere funzioni non presenti in $\text{\LaTeX}$
- ▶ modificare la definizione di alcuni comandi
- ▶ modificare il valore di alcuni parametri, come quelli di impaginazione

Sul sito principale distribuzione di $\text{\TeX}$ ci sono circa 6600 pacchetti, *Overleaf* ne ha più di 5000

Non è complicato aggiungere funzioni a $\text{\LaTeX}$ data la sua struttura modulare.

Il Preambolo - Esempio 1

Cambiare il preambolo spesso cambia del tutto il nostro documento

Preambolo

```
\documentclass{article}  
\usepackage{times}  
\usepackage[italian]{babel}  
\usepackage{lipsum}  
\begin{document}  
    \lipsum[30]  
\end{document}
```

I pacchetti - 2 - babel

```
\usepackage[< lingue >]{babel}
```

- ▶ Modifica vari nomi di “oggetti” del documento (Bibliografia, Capitolo, . . .), la sillabazione e aggiunge diverse funzionalità a seconda della lingua usata
- ▶ L'ultima che si specifica è la lingua base
- ▶ Copre circa 50 lingue (anche non in alfabeto latino), con varianti (l'inglese ne ha 5)
- ▶ se si usa solo “american english” non c'è bisogno di caricarlo
- ▶ se si usa solo l'italiano

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

- ▶ Se si usa il greco in un documento scritto in italiano si scriverà

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

Nel corso del documento si può scegliere di scrivere in greco

I pacchetti - 3

```
\usepackage{graphicx}
```

Serve per inserire immagini nel proprio documento di testo

```
\usepackage{amsmath}
```

Fornisce molte funzioni per scrivere testi di carattere matematico necessario se il documento contiene formule complesse

```
\usepackage{amssymb}
```

Permette di usare più simboli matematici rispetto a $\text{\LaTeX}$ base

I pacchetti - 4

```
\usepackage[T1]{fontenc}  
\usepackage[utf8]{inputenc}
```

sono “tecnici” e servono per la gestione dell’input e dell’output dei caratteri: vanno scritti per l’uso dell’italiano (e non solo)

Se qualcuno che ha il vostro file lo compila e vede caratteri sbagliati il problema è quasi certamente il pacchetto `inputenc` che è legato alla codifica dei caratteri del vostro sistema operativo e della vostra tastiera.

Senza quel pacchetto però usare le lettere accentate è più complicato.

Nel preambolo ci vanno anche altre cose

Il preambolo termina con

```
\begin{document}
```

Da qui inizia il documento vero e proprio che termina con

```
\end{document}
```

Tutto ciò che è scritto dopo viene ignorato

Alcuni pacchetti utili

Pacchetto

Funzioni

`geometry:`

è un pacchetto utile per ridefinire i margini delle pagine (alto - basso -sinistro - destro)

`enumerate ed`

consentono di creare elenchi puntati e numerati, o liste descrittive diverse da quelle standard, in modo semplice

`enumitem:`

consente di gestire intestazioni e piè di pagina (numerazione delle pagine, ecc.)

`fancyhdr:`

`xcolor:`

è un pacchetto per la gestione (di base e avanzata) dei colori

`beamer:`

è un pacchetto che consente di realizzare presentazioni (da proiettare) e/o lucidi (da stampare e proiettare)

`hyperref:`

serve a gestire i collegamenti ipertestuali nei file pdf creati da $\text{\TeX}$

`PSTricks, tikz`

permettono la realizzazione di grafici e disegni.

e pacchetti

collegati:

I caratteri - 1

Il testo in un documento $\text{\LaTeX}$ può essere completamente scritto utilizzando solo i caratteri con codici ASCII tra 32 e 126

! " # \$ % & ' () * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ 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 [\] ^ _
` 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 { | } ~

I caratteri - 2

Tutti i computer hanno a disposizione questi caratteri!

Quindi **tutti** i computer possono leggere, interpretare, compilare un file $\text{\LaTeX}$ scritto con questi caratteri.

Ma non tutti questi caratteri sono uguali . . .

I caratteri speciali - 1

In $\text{\LaTeX}$ esistono alcuni caratteri speciali

Carattere	Effetto	Comando
~	spazio insecabile	<code>\textasciitilde</code>
&	tabulazione	<code>\&</code>
#	argomento di funzioni	<code>\#</code>
_	indice nelle formule	<code>_</code>
\$	modo matematico	<code>\\$</code>
\	comando $\text{\LaTeX}$	<code>\textbackslash</code>
%	commento	<code>\%</code>
{	inizio gruppo	<code>\{</code>
^	esponente nelle formule	<code>\textasciicircum</code>
}	fine gruppo	<code>\}</code>

I Caratteri speciali - 2 (Esempio)

Scriviamo un documento con qualche carattere speciale

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}
\title{I caratteri speciali}
\begin{document}
\maketitle
Oggi ho iniziato a usare \LaTeX\ e ho scoperto che ci sono cose un po' strane.
Il prof.~Vassallo ci ha detto che non possiamo usare tutti i caratteri
perch\ 'e se ad esempio scrivo % dopo non si vede più niente \dots
```

Allora per scrivere 10 per cento devo scrivere 10\%

Però se devo scrivere Dürer non devo fare fatica basta scrivere D\"urer.

Poi è vero che per scrivere 100 dollari devo scrivere 100\\$, ma se devo scrivere x al quadrato mi basta fare x^2 .

```
\end{document}
```

Cambiare carattere

La famiglia di caratteri di default in $\text{\LaTeX}$ è *Computer Modern*

Serif Sans Serif Typewriter

Si può scegliere un pacchetto che modifichi la famiglia come, ad esempio, `lmodern`, `tgbonum`, `newpxtext` o `newpxtext`.

Un pacchetto per la scelta del carattere generalmente stabilisce tutte e tre le tipologie: ad esempio il pacchetto `newttext` fornisce

Times Helvetica Courier

A volte viene anche stabilito il carattere per la matematica, altrimenti bisogna caricare due diversi pacchetti (ad esempio `newpxtext` e `newpxmath`). Il pacchetto `mathdesign` è un valido aiuto in questi casi.

Non lanciatevi in esperimenti di accoppiamenti di caratteri: stabilire quali caratteri “stanno bene” insieme non è alla portata di tutti.

Cambiare carattere localmente

Se si vuole scegliere un carattere solo per una porzione di documento si usa la sintassi

```
{\fontfamily{xxx}\selectfont  
Questo testo ha un carattere diverso}
```

dove xxx è il codice del font.

```
{\fontfamily{put}\selectfont  
Questo testo usa il carattere Utopia}
```

Questo testo usa il carattere Utopia

Se si usano Lua $\text{\TeX}$ o X $\text{\TeX}$ si hanno a disposizione *tutti* i font OpenType presenti sul sistema.

Finire una riga

Abbiamo visto che andare a capo per $\text{\LaTeX}$ equivale a uno spazio, mentre andare a capo due volte inizia un nuovo capoverso

Si può andare a capo, senza interrompere il capoverso:

- ▶ con il comando `\newline`
- ▶ con il comando `\\[spazio]` che aggiunge uno spazio aggiuntivo (opzionale)

Invece il comando `\par` equivale alla riga vuota.

Finire una riga - Esempio

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

```
\usepackage[T1]{fontenc}
```

```
\usepackage[utf8]{inputenc}
```

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

```
\begin{document}
```

In questa riga andiamo a capo senza nessun comando e senza lasciare una riga bianca.

Frustrante, vero? Proviamo ad andare a capo due volte

Questo va, e c'è il rientro. Ho finito il paragrafo. Ora provo con un comando `\par`

Toh è come prima. Ora vado a capo a metà linea\\

Così va! E anche così `\newline`

E posso anche fare degli spazi (brutti e inutili)\\[4em]

Sì sta proprio male questo spazio!

```
\end{document}
```

Gli spazi

L^AT_EX inserisce automaticamente gli spazi orizzontali e verticali necessari.

Per forzare l'inserimento di spazi orizzontali possiamo scrivere

```
\hspace{lunghezza}  
\hspace*{lunghezza}
```

mentre per quelli verticali

```
\vspace{lunghezza}  
\vspace*{lunghezza}
```

La versione con asterisco impedisce che lo spazio venga eliminato all'inizio o fine riga o a fine pagina

Gli spazi - Esempio

```
Dobbiamo fare molto spazio:\hspace{2cm} ne abbiamo bisogno\\  
\hspace{3cm}Ora facciamo spazio perché ne abbiamo bisogno\\  
\hspace*{3cm}Ora facciamo spazio perché ne abbiamo bisogno  
\vspace{3em}\\\n  
Ne abbiamo bisogno in tutte le direzioni
```

Dobbiamo fare molto spazio:	ne abbiamo bisogno
Ora facciamo spazio perché ne abbiamo bisogno	
Ora facciamo spazio perché ne abbiamo bisogno	
Ne abbiamo bisogno in tutte le direzioni	

Gli spazi - Altri comandi

Altri comandi che possono servire per creare spazi sono

- ▶ `\quad` e `\qquad` che si usano soprattutto in ambito matematico che generano uno spazio orizzontale di `1em` e `2em`
- ▶ `\smallskip`, `\medskip` o `\bigskip` che generano uno spazio verticale di circa `3pt`, `6pt` e `12pt` circa.

I comandi - 1

In $\text{\LaTeX}$ qualsiasi operazione si voglia fare sul testo viene svolta mediante un **comando**

Comandi

Questo testo è `\textit{italico}`, mentre questo è `\textbf{grassetto}` è questo è `\texttt{a larghezza fissa}`.

Questo testo è *italico*, mentre questo è **grassetto** è questo è a larghezza fissa.

Posso anche scrivere `\small piccolo`, o `\footnotesize più piccolo` o `\tiny piccolissimo`

Posso anche scrivere piccolo, o più piccolo o piccolissimo

Ma si può anche `\large ingrandire` anche `\huge di più`

Ma si può anche ingrandire anche **di più**

I comandi - 2

Ci sono tre tipi di comandi, per quanto riguarda la forma:

- ▶ un singolo carattere
 - ▶ spazio _
 - ▶ tilde ~
 - ▶ circonflesso ^
 - ▶ linea bassa _
- ▶ barra rovescia e un carattere non alfabetico (simboli di controllo) come \\$
- ▶ barra rovescia e una successione di caratteri alfabetici (parole di controllo) come \textbf

Maiuscolo e minuscolo sono **diversi** nelle parole di controllo

Lo spazio

- ▶ più spazi contano come uno solo
- ▶ un fine riga conta come uno spazio
- ▶ una riga vuota conta come fine capoverso
- ▶ sa a fine riga scrivo %, allora non conta come spazio
- ▶ gli spazi dopo i comandi del terzo tipo sono **ignorati!**
infatti scriviamo `\LaTeX\` oppure `\LaTeX{\ }` e non `\LaTeX\`

Tilde

La tilde ~ indica a $\text{\LaTeX}$ di lasciare uno spazio che non può essere usato per andare a capo

tilde

Dante Alighieri scrisse ``La~Divina Commedia''

Dante Alighieri scrisse “La Divina Commedia”

Linea bassa e circonflesso

Si usano in matematica

Linea bassa

$$x^3 + x^2 - 1 \rightarrow x^3 + x^2 - 1$$

$$\log_2 8 = 3 \rightarrow \log_2 8 = 3$$

Simboli di controllo

Sono della forma “barra rovescia e un carattere non alfabetico” cioè un carattere che non è né una lettera né un numero

`\!` `\'` `\%` `\(` `\)` `\=` `\^` `\'` `\*` `\``
`\[` `\]` `\|` `\@` `\\` `\,` `\;` `\:` `\-` `\_`
`\_` `\&` `\/` `\$` `\{` `\}` `\.`

Alcuni li abbiamo già visti come `\'` o `\$`, altri li vedremo più avanti
Generalmente servono a modificare il carattere successivo o a scrivere caratteri non stampabili o a produrre spazi.

Comandi

I comandi possono essere

- ▶ dichiarazioni
- ▶ azioni

Dichiarazione

Un esempio è `\large` che ingrandisce il testo da quel punto in poi

Ingrandiamo `\large` i caratteri →

Ingrandiamo i caratteri

Azione

Un esempio è `\textbf` che fa scrivere una parte di testo in grassetto

Testo in `\textbf{grassetto}` → Testo in **grassetto**

Argomenti di un comando - 1

Spesso i comandi hanno un argomento, cioè un “qualcosa” (una porzione di testo, una lunghezza, ...) a cui viene applicato il comando.

Argomento

Nell'esempio di prima

- ▶ `\textbf` è il comando
- ▶ `grassetto` è l'argomento

Argomenti di un comando - 2

Possono esserci più argomenti

Due argomenti

```
\setlength{\fboxrule}{1pt}
```

- ▶ `\setlength` è il comando
- ▶ `\fboxrule` è il primo argomento
- ▶ `1pt` è il secondo argomento

```
\setlength{\fboxrule}{1pt} \fbox{a} → 
```

```
\setlength{\fboxrule}{3pt} \fbox{a} → 
```


Argomenti opzionali di un comando

Alcuni comandi hanno anche *argomenti opzionali* che vanno tra parentesi quadre

Ne abbiamo già visti:

Argomento opzionale

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

- ▶ `\usepackage` è il comando
- ▶ `italian` è l'argomento opzionale
- ▶ `babel` è l'argomento obbligatorio

Argomenti opzionali di un comando - Esempio

Un `\scalebox{2}{\fbox{esempio}}` → Un

esempio

Un `\scalebox{2}[.5]{\fbox{esempio}}` → Un

esempio

Un altro `\framebox{esempio}` → Un altro

esempio

Un altro `\framebox[3cm]{esempio}` → Un altro

esempio

Un altro `\framebox[3cm][r]{esempio}` → Un altro

esempio

Un altro `\framebox[3cm][l]{esempio}` → Un altro

esempio

Comandi per il testo: stile

Non è possibile fare un elenco completo dei comandi di $\text{\LaTeX}$, ma cominciamo a vedere quelli per il formato del testo

Comando	Dichiarazione	Nome	Effetto
<code>\textmd{testo}</code>	<code>\mdseries</code>	Serie media	testo
<code>\textbf{testo}</code>	<code>\bfseries</code>	Serie nera	testo
<code>\textrm{testo}</code>	<code>\rmfamily</code>	Tondo con grazie	testo
<code>\textsf{testo}</code>	<code>\sffamily</code>	Lineare senza grazie	testo
<code>\texttt{testo}</code>	<code>\ttfamily</code>	Monospaziato	testo
<code>\textup{testo}</code>	<code>\upshape</code>	Forma diritta	testo
<code>\textit{testo}</code>	<code>\itshape</code>	Forma corsiva	<i>testo</i>
<code>\textsl{testo}</code>	<code>\slshape</code>	Forma inclinata	<i>testo</i>
<code>\textsc{testo}</code>	<code>\scshape</code>	Maiuscoletto	TESTO
<code>\textnormal{testo}</code>	<code>\normalfont</code>	Font di default	testo

Comandi per il testo: grandezza

Per la grandezza del font sono a disposizione queste dichiarazioni (in ordine di grandezza)

```
\tiny      \scriptsize    \footnotesize    \small      \normalsize  
\large     \Large        \LARGE        \huge      \Huge
```

Sia queste dichiarazioni che le precedenti valgono **solo** in modalità testo e non in modalità matematica

Comandi per il testo - Esempio 1

```
\documentclass[a4paper]{article}
\usepackage[italian]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
Ci sono tanti modi per mettere in evidenza una parte di testo
\begin{itemize}
    \item \textbf{il grassetto}
    \item \textit{il corsivo}
    \item \textsl{l'obliquo}
    \item \textsc{il maiuscoletto}
\end{itemize}
È ritenuto cattivo stile \underline{sottolineare}. \\
\LaTeX ha anche una modalità ``intelligente'':
\verb|\emph{testo}| rende il testo corsivo se il testo in cui si
trova è tondo e lo rende tondo se il testo in cui si trova è
corsivo (come il testo di un teorema.\\
\itshape Questo testo è corsivo: una parola \emph{importante} in
tondo.\\
\normalfont Questo testo è tondo: una parola \emph{importante}
in corsivo.
\end{document}
```

Comandi per il testo - Esempio 2

```
\documentclass[a4paper]{article}  
\usepackage[italian]{babel}  
\usepackage[T1]{fontenc}  
\usepackage[utf8]{inputenc}  
\begin{document}
```

Possiamo ingrandire e rimpicciolire il testo, ma non conviene esagerare. \\

\emph{Dobbiamo ricordarci che i comandi per la grandezza dei font sono \textbf{dichiarazioni}.}\\

```
{\tiny tiny}  \\  {\scriptsize scriptsize}  \\  
  {\footnotesize footnotesize}  \\  {\small small}  \\  
    {\normalsize normalsize}\\  {\large large}  \\  
      {\Large Large}  \\  {\LARGE LARGE}  \\  
        {\huge huge}  \\  {\Huge Huge}  
\end{document}
```

Comandi per il testo: colore

Utilizzando il pacchetto `xcolor` è possibile anche colorare il testo.

La dichiarazione `\color{colore}` colora del colore indicato tutto ciò che segue

Il comando `\textcolor{colore}{testo}` rende `testo` di colore `colore`

`\textcolor{red}{rosso}` → **rosso**

Oltre ai colori fondamentali il pacchetto `xcolor` ha molti altri modelli di colore e ha la possibilità di mescolare i colori nelle percentuali volute.

Il testo - 1

Ripassare (o studiare!)

- ▶ Ortografia
- ▶ Uso degli accenti
- ▶ Sillabazione
- ▶ Punteggiatura
- ▶ Uso delle abbreviazioni, degli appellativi, delle maiuscole
- ▶ Uso delle unità di misura

Il testo - 2

L'unità “fondamentale” di un testo è il **periodo** (frase di senso compiuto che termina con un punto)

Uno o più periodi formano un **capoverso** (in $\text{\LaTeX}$ si separa andando a capo)

Si hanno poi le **sottosezioni**, le **sezioni** e, se il documento è grande i **capitoli** e, eventualmente, le **parti**.

Il testo - 2

Ci sono poi degli elementi “di contorno”, alcuni sempre presenti, altri solo in certi tipi di pubblicazioni

- ▶ Un titolo, con autore e altre informazioni
- ▶ Un indice
- ▶ Un'introduzione
- ▶ Una bibliografia
- ▶ Un indice analitico
- ▶ Eventuali appendici

$\text{\LaTeX}$ produce automaticamente indice, bibliografia, indice analitico

Suddivisione del testo

L^AT_EX ha a disposizione sette livelli di suddivisione del testo

- ▶ Parte `\part{titolo}`
- ▶ Capitolo `\chapter{titolo}` (non è definito nella classe `article`)
- ▶ Sezione (più correttamente in italiano “paragrafo”) `\section{titolo}`
- ▶ Sottosezione `\subsection{titolo}`
- ▶ Sottosottosezione `\subsubsection{titolo}`
- ▶ Paragrafo/capoverso (nessuna delle due dizioni è del tutto esatta) `\paragraph{titolo}`
- ▶ Sottoparagrafo `\subparagraph{titolo}`

Se al comando di sezionamento, ad esempio `\chapter` viene aggiunto un asterisco (`\chapter*{title}`), il capitolo non viene numerato e non viene riportato nell'indice.

Ambienti

Abbiamo visto che il testo di un documento $\text{\LaTeX}$ è racchiuso tra

`\begin{document}` e `\end{document}`

Questo è un esempio di una suddivisione logica di un documento che non corrisponde a un sezionamento: gli ambienti.

Alcuni esempi di questo tipo sono

- ▶ gli elenchi puntati e numerati
- ▶ le formule matematiche
- ▶ le tabelle
- ▶ le citazioni

Gruppi - 1

Possiamo raggruppare oggetti in $\text{\LaTeX}$ in vari modi:

- ▶ inserendo l'oggetto tra parentesi graffe `{ e }`
- ▶ inserendo l'oggetto tra `\begin{group e \end{group}`
- ▶ inserendo l'oggetto in un **ambiente**
- ▶ inserendo l'oggetto in una *scatola* `\fbox, \mbox,`
`\framebox, \makebox`

I gruppi possono essere annidati

È necessario rispettare l'ordine di annidamento

Ogni comando dato all'interno di un gruppo vale solo nel gruppo

Gruppi - 2

Il testo {\color{red} nel gruppo} è rosso



Il testo **nel gruppo** è rosso

Il metodo più corretto è però scrivere `\textcolor{red}{nel
gruppo}`

Gruppi - 3

Una scatola può servire a mantenere unita una parte di testo

`\mbox{I Promessi Sposi}` → I Promessi Sposi

Alcuni ambienti - 1

- ▶ **center**: consente di *centrare* intere porzioni di testo
`\begin{center}...\end{center}`
- ▶ **flushleft**: consente di *allineare a sinistra* intere porzioni di testo
`\begin{flushleft}...\end{flushleft}`
- ▶ **flushright**: consente di *allineare a destra* intere porzioni di testo
`\begin{flushright}...\end{flushright}`

Alcuni ambienti - 2

- ▶ **itemize:** crea elenchi puntati (il punto elenco predefinito è il pallino pieno; è possibile specificare il simbolo del punto elenco usando il pacchetto `enumitem`)
`\begin{itemize} \item... \end{itemize}`
- ▶ **enumerate:** crea elenchi numerati (la numerazione predefinita è in numeri arabi; usando i pacchetti `enumerate` o `enumitem` è possibile specificare che si desiderano numeri romani o lettere)
`\begin{enumerate} \item... \end{enumerate}`
- ▶ **description:** crea liste descrittive
`\begin{description} \item[etichetta]... \end{description}`

Alcuni ambienti - 3

- ▶ `tabular`: può essere usato per comporre tabelle, con o senza linee orizzontali e verticali che separano righe e colonne.

```
\begin{tabular}[opzioni] ... \end{tabular}
```

- ▶ `theorem`: per scrivere teoremi

```
\begin{theorem}[opzioni] ... \end{theorem}
```

Ambienti - Allineamento

Ci sono tre ambienti per posizionare il testo

- ▶ `center` (testo centrato)
- ▶ `flushright` (testo allineato a destra)
- ▶ `flushleft` (testo allineato a sinistra)

Tre dichiarazioni corrispondenti: `\centering`, `\raggedright` e `\raggedleft`.

È meglio **non** usare le dichiarazioni, ma gli ambienti (tranne la dichiarazione `\centering` all'interno degli ambienti `figure` e `table`).

Ambienti - Allineamento - Esempio

```
\documentclass[a4paper]{article}
\usepackage[italian]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
Testo normale
\begin{flushleft}
Questo testo è allineato a sinistra e non è giustificato quindi
l'aspetto non è bellissimo, ma è sicuramente più leggibile
di quello allineato a destra che è molto più difficile da
leggere.
\end{flushleft}
Testo normale
\begin{flushright}
Questo testo è allineato a destra e non è giustificato.
Non solo l'aspetto non è bellissimo, ma è più difficile da
leggere.
\end{flushright}
Testo normale
\begin{center}
Testo centrato
\end{center}
Testo normale
\end{document}
```

Citare testo - 1

Per citare del testo abbiamo tre ambienti:

- ▶ `quote`
- ▶ `quotation`
- ▶ `verse`

In tutti e tre gli ambienti il testo viene evidenziato mettendolo tra margini ristretti e modificando il carattere

Citare testo - 2

quote può contenere un solo capoverso

Il famosissimo \textit{incipit} dei \textit{Promessi Sposi} di
Alessandro Manzoni

\begin{quote}

Quel ramo del lago di Como, che volge a mezzogiorno, tra due
catene non interrotte di monti, tutto a seni e a golfi, \dots

\end{quote}

descrive in modo mirabile il ramo lecchese del Lago di Como (o
Lario). Gli altri rami sono quello di Como e quello di Gera.

Il famosissimo *incipit* dei *Promessi Sposi* di Alessandro Manzoni
*Quel ramo del lago di Como, che volge a mezzogiorno, tra
due catene non interrotte di monti, tutto a seni e a golfi, ...*

descrive in modo mirabile il ramo lecchese del Lago di Como (o Lario).
Gli altri rami sono quello di Como e quello di Gera.

Citare testo - 3

quotation può contenere più capoversi

Il Vangelo di Giovanni è notevolmente diverso dagli altri tre vangeli, detti sinottici. Già il suo inizio

```
\begin{quotation}
```

In principio era il Verbo, il Verbo era presso Dio e il Verbo era Dio.

Egli era in principio presso Dio: tutto è stato fatto per mezzo di lui, e senza di lui niente è stato fatto di tutto ciò che esiste.

```
\end{quotation}
```

indica che l'autore vuole approfondire la questione dell'identità del Cristo, inserendo ampie digressioni teologiche.

Il Vangelo di Giovanni è notevolmente diverso dagli altri tre vangeli, detti sinottici.

Già il suo inizio

In principio era il Verbo, il Verbo era presso Dio e il Verbo era Dio.

Egli era in principio presso Dio: tutto è stato fatto per mezzo di lui, e senza di lui niente è stato fatto di tutto ciò che esiste.

indica che l'autore vuole approfondire la questione dell'identità del Cristo.

Citare testo - 4

verse serve per citare/scrivere poesie: alla fine di ogni verso bisogna indicare di andare a capo con il comando \\ e ogni strofa è trattata come un capoverso, cioè deve essere lasciata una riga bianca fra una strofa e la successiva

```
\begin{verse}  
Nel mezzo del cammin di nostra vita\\  
mi ritrovai per una selva oscura,\\  
ché la diritta via era smarrita.  
  
Ahi quanto a dir qual era è cosa dura\\  
esta selva selvaggia e aspra e forte\\  
che nel pensier rinova la paura!  
\end{verse}
```

*Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura,
ché la diritta via era smarrita.*

*Ahi quanto a dir qual era è cosa dura
esta selva selvaggia e aspra e forte
che nel pensier rinova la paura!*

Ambienti - Elenchi - Esempio 1

```
\documentclass[a4paper]{article}
\usepackage[italian]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
Il mio primo elenco puntato
\begin{itemize}
\item primo
\item secondo
\item terzo
\end{itemize}
Il mio primo elenco numerato
\begin{enumerate}
\item Uno
\item Due
\item Tre
\end{enumerate}
Il mio primo elenco descrittivo
\begin{description}
\item[primo item] primo punto
\item[secondo item] secondo punto
\item[terzo item] terzo punto
\end{description}
```

```
Gli elenchi si possono annidare
\begin{enumerate}
\item primo punto dell'elenco
\item secondo punto dell'elenco
\begin{enumerate}
\item primo punto del sottoelenco
\item secondo punto del sottoelenco
\begin{itemize}
\item ancora più annidato
\item questo elenco
\end{itemize}
\item terzo punto del
sottoelenco
\end{enumerate}
\item terzo punto dell'elenco
\end{enumerate}
\end{document}
```

Ambienti - Elenchi - Esempio 2

```
\documentclass[a4paper]{article}
\usepackage[italian]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
Elenchi puntati
\begin{itemize}
\item primo
\begin{itemize}
\item primo del secondo livello
\item secondo del secondo livello
\begin{itemize}
\item primo del terzo livello
\item secondo del terzo livello
\end{itemize}
\item terzo del secondo livello
\end{itemize}
\item secondo
\item terzo
\end{itemize}
```

```
Elenchi numerati
\begin{enumerate}
\item primo
\begin{enumerate}
\item primo del secondo livello
\item secondo del secondo livello
\begin{enumerate}
\item primo del terzo livello
\item secondo del terzo
livello
\end{enumerate}
\end{enumerate}
\item terzo del secondo livello
\end{enumerate}
\item secondo
\item terzo
\end{enumerate}
\end{document}
```

Ambienti - Elenchi - 1

È facile modificare i contrassegni degli elementi di un elenco usando uno dei pacchetti

- ▶ `enumerate`
- ▶ `enumitem`

Ambienti - Elenchi - 2

Ad esempio con `enumitem` scrivendo

```
\begin{enumerate}[label=\textbf{Esercizio \arabic*}]  
  \item 2+2  
  \item 3+3  
\end{enumerate}
```

otteniamo

Esercizio 1 2+2

Esercizio 2 3+3

Ambienti - Tabelle - 1

Esempio

```
\begin{tabular}[allineamento]{colonne}  
    ... & ... & ... \\  
    ... & ... & ... \\  
\end{tabular}
```

allineamento può essere t, b, c

Ambienti - Tabelle - 2

Esempio

```
\begin{tabular}[allineamento]{colonne}  
  ... & ... & ... \\  
  ... & ... & ... \\  
\end{tabular}
```

colonne sono i descrittori dell'allineamento del testo nelle colonne (uno per colonna)

- ▶ l: colonna allineata a sinistra,
- ▶ c: colonna centrata,
- ▶ r: colonna allineata a destra,
- ▶ p{larghezza}: colonna composta da un breve capoverso.

Tra i descrittori di colonna vanno i separatori: | per un filetto verticale (da evitare!).

Si può scrivere `*{numero}{descrittore}` per ripetere numero di volte un descrittore.

Ambienti - Tabelle - 3

Esempio

```
\begin{tabular}[allineamento]{colonne}  
  ... & ... & ... \\  
  ... & ... & ... \\  
\end{tabular}
```

La tabella si compone riga per riga:

- ▶ le celle sono separate da &
- ▶ il termine della riga si indica con \\
- ▶ con il comando `\hline` si inserisce una riga orizzontale **può essere solo**
 - ▶ primo elemento di una tabella,
 - ▶ dopo il comando di fine riga \\
 - ▶ dopo un altro comando `\hline` (da evitare!).

Ambienti - Tabelle - 4

Il comando

```
\multicolumn{numero}{posizione}{cella}
```

serve per comporre un'unica cella con la posizione indicata, che si espande per `numero` colonne.

Il comando

```
\cline{col1 -col2}
```

serve per tracciare una linea orizzontale tra le colonne `col1` e `col2`, non si possono raddoppiare le linee, ma si possono avere due linee su due intervalli di colonne diverse.

Ambienti - Tabelle - Esempi

Esempio 1

```
\begin{tabular}{lcr}  
\hline  
Squadra & Città & Maglia \\  
\hline  
Inter & Milano & Nerazzurra  
\\  
Juventus & Torino &  
Bianconera \\  
Fiorentina & Firenze & Viola  
\\  
\hline  
\end{tabular}
```

Squadra	Città	Maglia
Inter	Milano	Nerazzurra
Juventus	Torino	Bianconera
Fiorentina	Firenze	Viola

Ambienti - Tabelle - Esempi

Esempio 2

```
\begin{tabular}{l*{4}{c}}  
\hline  
Lettera &  $\alpha$  &  $\beta$  &  $\gamma$  &  $\delta$  & \\  
Nome & alfa & beta & gamma & delta & \\  
\hline  
\end{tabular}
```

Lettera	α	β	γ	δ
Nome	alfa	beta	gamma	delta

Ambienti - Tabelle - Esempi

Esempio 3

```
\begin{tabular}{lp{0.7\textwidth}}
\hline
\textbf{Monarchia}&
La monarchia, letteralmente "Governo di uno" è una forma di
governo
in cui la carica di capo di stato è ricoperta da un re o da una
regina oppure un qualsiasi altro sovrano del ceto nobiliare. \\
\textbf{Democrazia}&
Democrazia etimologicamente significa ``governo del popolo'',
ovvero forma di governo in cui la sovranità è esercitata,
direttamente o
indirettamente, dal popolo, che generalmente è identificato come
l'insieme dei cittadini che ricorrono, in generale, a strumenti
di consultazione popolare.\\
\hline
\end{tabular}
```

Ambienti - Tabelle - Esempi

Esempio 3

Monarchia	La monarchia, letteralmente “Governo di uno” è una forma di governo in cui la carica di capo di stato è ricoperta da un re o da una regina oppure un qualsiasi altro sovrano del nobiliare.
Democrazia	La democrazia etimologicamente significa “governo del popolo”, ovvero forma di governo in cui la sovranità è esercitata, direttamente o indirettamente, dal popolo, che generalmente è identificato come l'insieme dei cittadini che ricorrono in generale a strumenti di consultazione popolare.

Ambienti - Tabelle - Esempi

Esempio 4

```
\begin{tabular}{llc}
\multicolumn{3}{c}{Menù}\\
\hline
\multicolumn{2}{c}{Piatto}
& Prezzo\\
\cline{1-2}
Tipo & Descrizione & \\
\hline
Pasta & al pomodoro & 6,50\\
& al pesto & 8,00\\
Bistecca & di manzo & 14,00\\
Crostata & & 6,00\\
\hline
\end{tabular}
```

Menù		
Piatto		Prezzo
Tipo	Descrizione	
Pasta	al pomodoro	6,50
	al pesto	8,00
Bistecca	di manzo	14,00
Crostata		6,00

Ambienti - Tabelle - Pacchetti utili

- ▶ `booktabs`: migliora l'aspetto grafico;
- ▶ `array`: estende gli ambienti `array` e `tabular` permettendo di inserire comandi validi per tutte le colonne e definire nuovi tipi di colonna;
- ▶ `tabularx`: permette di avere colonne di tipo `p` la cui larghezza è calcolata da $\text{\LaTeX}$;
- ▶ `multirow`: per unire più celle su una stessa colonna
- ▶ `colortbl`: per tabelle colorate;
- ▶ `longtable`: per tabelle molto lunghe che si estendono su più pagine

Le note

$\text{\LaTeX}$ permette di scrivere note

- ▶ a piè di pagina
- ▶ marginali

Per scrivere note al termine del capitolo o del libro bisogna usare il pacchetto `endnotes`

Le note a piè di pagina

Per scrivere una nota a piè di pagina si usa il comando `\footnote`

```
\footnote[richiamo]{testo della nota}
```

Il richiamo di default (che quindi non va scritto) è un numero all'apice.

Può essere anche un asterisco o altro.

Nella nota non possono essere inseriti tutti i comandi.

All'occorrenza si può fare uso di un comando separato per inserire il richiamo e per scrivere il testo:

```
\footnotemark[richiamo]  
\footnotetext[richiamo]{testo della nota}
```

Le note marginali

Se i margini del documento sono abbastanza ampi, possono esserci le *note a margine*. Il comando per produrle è `\marginpar` e segue la sintassi:

```
\marginpar[nota di sinistra]{nota di destra}
```

Figure in L^AT_EX

È necessario caricare il pacchetto `graphicx` per interpretare l'immagine e inserirla nel documento.

Inclusione della figura

1. importando un file grafico esterno, in un formato accettato da L^AT_EX ;
2. “disegnando” direttamente all'interno del documento utilizzando uno dei pacchetti dedicati: `tikz`, `PSTricks`, `pgfplots`, `xypic`, `mfpic`, ecc.
3. usando un pacchetto che funzioni da interfaccia verso un programma esterno come `asymptote` o `gnuplot`.

Formati grafici

- ▶ *immagini vettoriali*: descritte da forme, possono essere scalate e deformate senza perdere di definizione; i formati più usati sono pdf, ps (e il suo parente eps), svg;
- ▶ *immagini bitmap*: sono matrici di pixel colorati, di solito perdono in definizione se ingrandite o rimpicciolite; i formati più noti sono jpeg, png, bmp e spesso sono compresse.

Formati grafici in $\text{\LaTeX}$

- ▶ $\text{pdf\LaTeX}$ gestisce file in formato jpg, png, pdf, eps
- ▶ il pacchetto `bmpsize` permette inoltre di inserire file bitmap di altri formati come bmp e gif

Se il file grafico non è supportato da $\text{\LaTeX}$ bisogna convertirlo

Inclusione di un'immagine

L'immagine viene inclusa in un documento utilizzando il comando:

```
\includegraphics[opzioni]{nomefile}
```

Le opzioni sono

scale=fattore di scala consente di scalare proporzionalmente il grafico;

width=unità consente di specificare la larghezza del grafico;

height=unità consente di specificare l'altezza del grafico;

keepaspectratio=true/false evita di distorcere le immagini se vengono specificati sia l'altezza che la larghezza, che diventano ora un limite da non superare;

angle=angolo di rotazione consente di specificare l'angolo di rotazione della figura, in gradi ed in senso antiorario.

...

Programmi per la grafica

Per generare le immagini da inserire ricordiamo il programma *Geogebra* che permette di generare facilmente e in modo grafico sia figure geometriche che grafici di funzioni di una e due variabili ed esportarli in formati compatibili con $\text{\LaTeX}$.

Un altro programma che è in grado di generare immagini vettoriali è *Inkscape*.

Scatole - 1

Una **scatola** è un oggetto che può contenere del testo o una figura
Il suo contenuto viene trattato come se fosse un oggetto unico (non si spezza a fine riga).

```
\mbox{testo}  
\makebox[larghezza][posizione]{testo}  
\fbox{testo}  
\framebox[larghezza][posizione]{testo}
```

Il testo occuperà una sola riga.

Scatole - 2

I comandi che cominciano con “f” inseriscono una cornice attorno al testo; quelli che cominciano con la lettera “m” no.

Se la `larghezza` (facoltativa) viene indicata, il testo viene inserito dentro una scatola della larghezza specificata, Se viene specificata la `larghezza` allora si può specificare anche la `posizione`: “l” per allineare il testo a sinistra, “r” per allineare il testo a destra, “s” per distribuire il testo, allargando o restringendo lo spazio tra le parole, altrimenti il testo è centrato.

Scatole - Esempi

```
\fbox{testo}\  
\mbox{testo}\  
\makebox[50mm]{testo  
centrato}\  
\framebox[50mm][l]{testo a  
sinistra}\  
\framebox[50mm][r]{testo a  
destra}\  
\framebox[50mm]{testo  
centrato}\  
\framebox[50mm][s]{testo  
spaziato}\  

```

testo

testo

testo centrato

testo a sinistra

testo a destra

testo centrato

testo

spaziato

Scatole con più righe

comando `parbox` e ambiente `minipage`:

```
\parbox[allineamento][altezza][pos.interna]{larghezza}{testo}
```

oppure

```
\begin{minipage}[allineamento][altezza][pos.interna]{larghezza}  
testo  
\end{minipage}
```

`larghezza` rappresenta la larghezza della scatola.

Scatole con più righe - Esempi

```
\fbox{
\begin{minipage}{.75\linewidth}
Questo testo è in una \texttt{minipage.}\\
Posso andare a capo.\\
Si possono anche scrivere delle note\footnote{Questa è una
nota}.
\end{minipage}
}
```

Questo testo è in una minipage.

Posso andare a capo.

Si possono anche scrivere delle note^a.

^aQuesta è una nota

Scatole con più righe - Esempi

```
\fbox{
\begin{minipage}{.25\linewidth}
Questo testo è in una \texttt{minipage} molto più stretta.\\
Posso andare a capo.\\
Si possono anche scrivere delle note\footnote{Questa è una
nota}.
\end{minipage}
}
```

Questo testo è in
una minipage
molto più stretta.
Posso andare a
capo.
Si possono an-
che scrivere delle
note^a.

^aQuesta è una
nota

Lunghezze assolute - 1

Quando inserisco una figura specificando la larghezza (o l'altezza) devo darle la misura

$\text{\LaTeX}$ riconosce alcune unità di misura (alcune un po' "strane")

- ▶ `cm` centimetri.
- ▶ `mm` millimetri.
- ▶ `in` pollici.
- ▶ `pt` punti tipografici ($1\text{ pt} = (1/72,27)\text{ in} = 0,3515\text{ mm}$).
- ▶ `ex` x-height, altezza della lettera "x".
- ▶ `em` em-width, larghezza della "M",
- ▶ e poi ancora `pc` pica ($1\text{ pc} = 12\text{ pt}$), `bp` punto PostScript o big point ($1\text{ bp} = (1/72)\text{ in} = 0,3528\text{ mm}$), `sp` scaled point (la frazione 2^{-16} di un punto tipografico), `dd` punto didot ($1\text{ dd}=0,3760\text{ mm}=1,070\text{ pt}$), `cc` cicero ($1\text{ cc}=12\text{ dd}$)

Lunghezze assolute - 2

Quindi posso scrivere

```
\includegraphics[width=4cm]{immagine}  
\includegraphics[width=40mm]{immagine}  
\includegraphics[width=1.3in]{immagine}
```

Lunghezze relative - 1

Che problema c'è?

Se io decido di mettere il testo su due colonne la mia figura non ci sta più

Se io riporto la figura su una slide è piccola

Allora è meglio utilizzare delle lunghezze **relative**, cioè determinate dalle lunghezze della pagina

- ▶ `\linewidth` = lunghezza della riga di stampa (differisce se su una o più colonne)
- ▶ `\textwidth` = larghezza della gabbia di testo
- ▶ `\columnwidth` = larghezza della colonna
- ▶ `\textheight` = altezza della gabbia di testo

Figure - Esempio 1

```
\includegraphics{lion3}
```



Figure - Esempio 2 (scale)

```
\includegraphics[scale=.3]{lion3}
```



Figure - Esempio 3 (width e height)

```
\includegraphics[width=1cm]{lion3}
```



```
\includegraphics[height=2cm]{lion3}
```



Figure - Esempio 4 (angle)

```
\includegraphics[angle=90]{lion3}
```



Figure - Esempio 5 (ordine opzioni)

```
\includegraphics[angle=30,width=3cm]{lion3},
```



```
\includegraphics[width=3cm,angle=30]{lion3}
```



Gli stili della pagina - 1

- ▶ `empty`: testatina e piè di pagina vuoti
- ▶ `plain`: testatina vuota, piè di pagina con numero di pagina al centro (`article` e `report`)
- ▶ `headings`: piè di pagina vuoto, testatina sinistra numero di pagina a sinistra capitolo a destra, testatina destra paragrafo a sinistra numero di pagina a destra (`book`)
- ▶ `myheadings`: come `headings`, ma gli spazi per capitolo e paragrafo lasciati a scelta dell'autore

Gli stili della pagina - 2

Si usa il pacchetto `fancyhdr`

Si sceglie con `\pagestyle{stile}` nel preambolo

È possibile modificare lo stile per una pagina con il comando

`\thispagestyle{stile}`.

La geometria della pagina

La geometria della pagina si gestisce col pacchetto `geometry` che permette di regolare in modo molto fine margini, lunghezza di riga, distanza tra testatina e testo,

Da usare con cautela: il disastro estetico è dietro l'angolo!

Scrivere “verbatim” - 1

Come posso fare a far scrivere a $\text{\LaTeX}$ `\textheight` senza ottenere un errore?

E, più in generale, come faccio a inserire del codice di un programma e, in particolare, del codice $\text{\LaTeX}$?

Inserire del testo senza che $\text{\LaTeX}$ lo interpreti, cioè **verbatim**.

Molti pacchetti: `listing`, `fancyvrb` e `fvextra`

Scrivere “verbatim” - 2

Inserimento del testo *verbatim* di base.

Inserimento in linea di poche parole

```
\verb simbolo testo da riprodurre simbolo  
\verb* simbolo testo da riprodurre simbolo
```

“simbolo” è un solo carattere e delimita il testo da riprodurre (“+” o “|”); la versione senza asterisco riproduce gli spazi come spazi; la versione con l’asterisco riproduce gli spazi con il segno speciale `_`.

Se scriviamo `\verb+\textcolor{red}{casa}+` ci apparirà il comando $\text{\LaTeX}$ e non la parola “casa” in rosso.

Normalmente il testo è riprodotto in `typewriter`.

Scrivere “verbatim” - 3

L'ambiente è invece

```
\begin{verbatim}  
testo da riprodurre  
di solito  
su più righe  
\end{verbatim}
```

oppure

```
\begin{verbatim*}  
...  
\end{verbatim*}
```

Un problema

Vogliamo mandare il nostro file tex a un nostro amico negli Stati Uniti che ha una tastiera senza lettere accentate

Quindi non possiamo usare direttamente “è”, “à”, ecc. ma dobbiamo usare i comandi.

Scrivere “é” non è difficile \ ' e

Scrivere “è” è più lungo perché il simbolo ` non c'è sulla tastiera.

Una soluzione

Scriveremo *nel preambolo*

```
\newcommand{\aacc}{à}
\newcommand{\eacc}{è}
\newcommand{\iacc}{ì}
\newcommand{\oacc}{ò}
\newcommand{\uacc}{ù}
```

ed ora scrivere “à” è immediato. . . basta scrivere `\aacc`

Perché non abbiamo usato `\a`? Cosa è quello strano comando per la “i”?

Nuovi comandi - 1

I vettori si indicano in grassetto

$\text{\texttt{\$}\texttt{\backslash mathbf}\texttt{\{v}\}\texttt{\$}} \rightarrow \mathbf{v}$

ma c'è chi preferisce avere una freccia sopra

$\text{\texttt{\$}\texttt{\overrightarrow}\texttt{\{v}\}\texttt{\$}} \rightarrow \overrightarrow{v}$

Cosa succede se vogliamo (o dobbiamo) cambiare notazione?

Nuovi comandi - 2

Definiamo un comando `\vect[1]` in questo modo

```
\newcommand{\vect}[1]{\mathbf{#1}}
```

Cosa vuol dire?

```
\newcommand{nome comando}[num. argomenti]{definizio
```

- ▶ `nome comando` è il nome del nuovo comando;
- ▶ `num. argomenti` è un numero, il numero di argomenti della funzione;
- ▶ `definizione` è il testo o la sequenza di comandi che deve essere sostituito tutte le volte che si scrive `\nuovo comando`; un parametro della forma `#n` sarà sostituito dall'ennesimo argomento della funzione.

Nuovi comandi - 3

```
\newcommand{\vect}[1]{\mathbf{#1}}
```

vuol dire “ogni volta che scrivo `\verb{v}` traducilo come `\mathbf{v}`”

x

Se cambiamo idea e vogliamo la freccia sopra?

Cambio il comando in

```
\newcommand{\vect}[1]{\overrightarrow{#1}}
```

e automaticamente tutti i vettori avranno la freccia sopra

$\overrightarrow{x}$

Nuovi comandi - 4

Nel caso precedente il comando ha un solo argomento (“x” nell’esempio).

Vediamo adesso un esempio di comando con due argomenti:

```
\newcommand{\Sum}[2]{\left(#1 + #2 = \text{the}\numexpr #1 + #2\right) \relax \}
```

Questo comando prendo i numeri #1 e #2 e scrive (in modalità matematica)

#1+#2 = (la somma di #1 e #2)

`\Sum{2}{3}` → $2+3=5$

Nuovi comandi - 5

Se il comando è già definito (da $\text{\LaTeX}$ o da noi) dobbiamo cambiare sintassi

```
\renewcommand{nomecomando}[argomenti]{definizione}
```

Si possono creare anche nuovi ambienti o ridefinire ambienti già esistenti:

```
\newenvironment{nome}[argomenti]  
{definizione inizio} {definizione fine}
```

- ▶ `nome` è il nome dell'ambiente,
- ▶ `argomenti` è il numero dei parametri presenti nella definizione,
- ▶ `definizione inizio` è il testo che verrà sostituito ogni volta che scriveremo `\begin{nome}` e `definizione fine` è il testo che verrà sostituito ogni volta che scriveremo `\end{nome}`.

L^AT_EX è nato per scrivere **bene** la matematica.

Due modalità matematiche

- ▶ *l'ambiente `math`*: utilizzato per le formule che devono comparire all'interno del testo;
- ▶ *l'ambiente `displaymath`*: utilizzato per le formule che stanno su una linea a sé stante.

Molti editor hanno barre degli strumenti che facilitano l'inserimento dei vari simboli.

L'ambiente `math`

Due comandi

► `\ ( . . . \)` (preferibile)

► `$ . . . $`

`\ (2x-y=3\)`

`$2x-y=3$`

`2x-y=3`

`2\textit{x}-\textit{y}=3`

$2x - y = 3$	$2x - y = 3$	$2x - y = 3$	$2x - y = 3$
--------------	--------------	--------------	--------------

L'ambiente `displaymath`

Due comandi per creare l'ambiente (e altri per equazioni numerate):

► `\[ \dots \]` (molto preferibile)

► `$$ \dots $$`

`\[2x-y=3\]`

`$$2x-y=3$$`

il risultato sarà sempre:

$$2x + y = 3$$

Gli spazi non hanno alcun effetto quando si scrive in modalità matematica

$$2x+y=3$$

e

$$2x+y=3$$

danno lo stesso risultato.

Nell'ambiente `displaymath` non si devono lasciare linee bianche

Differenze tra gli ambienti `math` e `displaymath`

Tra i due ambienti cambia, oltre alla posizione del testo, la grandezza di alcuni operatori e la posizione di indici e pedici

```
\( \lim_{x\to 0}\frac{\sen x}{x}=1 \quad \int_0^1 x^2\,dx=\frac{1}{3} \quad \sum_{n=0}^{+\infty}q^n=\frac{1}{1-q} \)
```

$$\lim_{x\rightarrow 0} \frac{\sen x}{x} = 1 \qquad \int_0^1 x^2 dx = \frac{1}{3} \qquad \sum_{n=0}^{+\infty} q^n = \frac{1}{1-q}$$

$$\lim_{x\rightarrow 0} \frac{\sen x}{x} = 1 \qquad \int_0^1 x^2 dx = \frac{1}{3} \qquad \sum_{n=0}^{+\infty} q^n = \frac{1}{1-q}$$

Alcuni operatori matematici

Operatore	Comando	Esempio	Risultato
Apici	<code>^{\}</code>	<code>\$x^{2}\$</code>	x^2
Pedici	<code>_{\}</code>	<code>\$x_{1}\$</code>	x_1
Frazioni	<code>\frac{num}{den}</code>	<code>\$\frac{1}{2}\$</code>	$\frac{1}{2}$
Radici	<code>\sqrt[indice]{argomento}</code>	<code>\$\sqrt[5]{x}\$</code>	$\sqrt[5]{x}$
Sommatorie	<code>\sum_{ind. inf.}^{ind. sup.}arg</code>	<code>\$\sum_{n=0}^5q^n\$</code>	$\sum_{n=0}^5 q^n$
Integrali	<code>\int_{inf.}^{sup.}f(x)dx</code>	<code>\$\int_0^33x^4\,dx\$</code>	$\int_0^3 3x^4 dx$

Alcune funzioni

<code>\arccos</code>	<code>arccos</code>	<code>\exp</code>	<code>exp</code>	<code>\min</code>	<code>min</code>	<code>\arcsin</code>	<code>arcsin</code>
<code>\gcd</code>	<code>gcd</code>	<code>\lg</code>	<code>lg</code>	<code>\ln</code>	<code>ln</code>	<code>\sup</code>	<code>sup</code>
<code>\arctan</code>	<code>arctan</code>	<code>\det</code>	<code>det</code>	<code>\lim</code>	<code>lim</code>	<code>\log</code>	<code>log</code>
<code>\tan</code>	<code>tan</code>	<code>\inf</code>	<code>inf</code>	<code>\max</code>	<code>max</code>	<code>\sin</code>	<code>sin</code>

Per avere le funzioni “in italiano” bisogna definirle. Usando il pacchetto `amsmath`

```
\DeclareMathOperator{comando}{testo che appare}
```

```
\DeclareMathOperator{\sen}{sen}
```

Lettere greche

Minuscole

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

Maiuscole

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

Altri simboli

Simbolo	Comando	Esempio	Risultato
Appartenenza	<code>\in</code>	<code>\$5\in \mathbb{N}\$</code>	$5 \in \mathbb{N}$
Esiste	<code>\$_exists\$</code>	<code>&\$\exists n > 3\$</code>	$\exists n > 3$
Per ogni	<code>\$_forall\$</code>	<code>\$_forall x \in \mathbb{R}\$</code>	$\forall x \in \mathbb{R}$
Freccia sopra	<code>\$_vec{}\$</code>	<code>\$_vec{v}\$</code>	$\vec{v}$
Sottoinsieme	<code>\$_subset\$</code>	<code>\$A\subset B\$</code>	$A \subset B$
Unione	<code>\$_bigcup\$</code>	<code>\$A\bigcup B\$</code>	$A \cup B$
Intersezione	<code>\$_bigcap\$</code>	<code>\$A\bigcap B\$</code>	$A \cap B$
Implicazione	<code>\$_Rightarrow\$</code>	<code>\$p\Rightarrow q\$</code>	$p \Rightarrow q$
Puntini	<code>\$_dots\$</code>	<code>\$1, 2, 3, \dots\$</code>	$1, 2, 3, \dots$
Puntini	<code>\$_dotsb\$</code>	<code>\$1+ 2+ \dotsb + n\$</code>	$1 + 2 + \dots$
Minore o uguale	<code>\$_leq\$</code>	<code>\$3 \leq 5\$</code>	$3 \leq 5$
Maggiore o uguale	<code>\$_geq\$</code>	<code>\$8 \geq 5\$</code>	$8 \geq 5$
Minore o uguale	<code>\$_leq\$</code>	<code>\$3 \leq 5\$</code>	$3 \leq 5$
Diverso	<code>\$_neq\$</code>	<code>\$8 \neq 5\$</code>	$8 \neq 5$

Parentesi - 1

Scrivere `\[(\frac{1}{2})^2\]` produce il pessimo risultato

$$(\frac{1}{2})^2$$

Per avere parentesi più adatte ci sono due strade:

- ▶ parentesi “espandibili”
- ▶ parentesi più grandi (col pacchetto `amsmath`)

La prima strada è la più semplice, ma a volte è necessario usare l'altra.

Parentesi - 2

Le parentesi espandibili hanno sintassi

`\left( \right)` `\left[ \right]`

`\left\{ \right\}`

e producono

$$\left(\frac{1}{2}\right) \quad \left[\frac{1}{2}\right] \quad \left\{\frac{1}{2}\right\}$$

Ci sono altri tipi di parentesi espandibili come

`$$\left\langle \right\rangle\right.$` che produce $\langle \rangle$: è importante la coppia `$$\left. \right.\right.$` che produce parentesi “invisibili” necessarie perché $\text{\LaTeX}$ controlla sempre che i delimitatori siano bilanciati, cioè a ogni parentesi aperta con un `\left` ne corrisponda una chiusa con `\right`

Parentesi - 3

I delimitatori di grandezza fissa sono

`\Biggl(\biggl(\Bigl(\bigl(())\bigr)\Bigr)\biggr)\Big`
(e gli analoghi per le altre parentesi) e producono

$$\left(\left(\left((\right)\right)\right)\right)$$

Testo all'interno di una formula

Come faccio a scrivere

$$f(x) > 1 \text{ se } x < 3$$

A volte può essere necessario inserire del testo in una formula.

Se scrivo

`\[f(x) > 1 se x < 3\]`

ottengo

$$f(x) > 1sex < 3$$

Devo tornare al modo testo!

Il comando da usare è `\text{testo}`

`\[f(x) > 1 \text{ se } x < 3\]`

Gli spazi prima e dopo la parola “se” sono indispensabili: la spaziatura nel modo matematico è generata automaticamente e non dipende dagli spazi che scrivo.

Cambiare i font - 1

I comandi: `\mathit`; `\mathrm`; `\mathbf` e `\mathsf` permettono di cambiare lo stile dei caratteri in modalità matematica.

Occorre però ricordare che *non tutti i caratteri matematici sono disponibili con tutti gli stili*.

Il comando `\mathcal` genera delle lettere maiuscole corsive. Ad esempio: `$\mathcal{C}$` fa stampare: $\mathcal{C}$.

I simboli per gli insiemi numerici $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{N}$, ecc. si ottengono con il comando `$\mathbb{R}$` , ecc. (`\mathbb`= blackboard bold).

Cambiare i font - 2

I comandi: `\mathit`; `\mathrm`; `\mathbf` e `\mathsf` permettono di cambiare lo stile dei caratteri in modalità matematica.

Occorre però ricordare che *non tutti i caratteri matematici sono disponibili con tutti gli stili*.

Il comando `\mathcal` genera delle lettere maiuscole corsive. Ad esempio: `$\mathcal{C}$` fa stampare: $\mathcal{C}$.

I simboli per gli insiemi numerici $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{N}$, ecc. si ottengono con il comando `$\mathbb{R}$` , ecc. (`\mathbb`= blackboard bold).

Ambienti matematici 1 - Array

Le matrici (o tutto ciò che può essere scritto come una tabella) si possono ottenere con l'ambiente `array` (simile a `tabular`).

```
\[{\mathcal A} = \left(
\begin{array}{ccc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}
\right)
```

$$\mathcal{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

I parametri usati sono come quelli di `tabular`.

Ambienti matematici 2 - Matrici

Il pacchetto `amsmath` mette a disposizione altri ambienti per le matrici

- ▶ `\begin{pmatrix}...\end{pmatrix}` → $(matrice)$
- ▶ `\begin{bmatrix}...\end{bmatrix}` → $[matrice]$
- ▶ `\begin{Bmatrix}...\end{Bmatrix}` → $\{matrice\}$
- ▶ `\begin{vmatrix}...\end{vmatrix}` → $|matrice|$
- ▶ `\begin{Vmatrix}...\end{Vmatrix}` → $\|matrice\|$
- ▶ `\begin{matrix}...\end{matrix}` → $matrice$

La spaziatura è inferiore a quella di `array`. Al massimo 10 colonne centrate: se le colonne devono essere allineate a destra o a sinistra, bisogna usare `array`.

L'ambiente `cases` si usa per le funzioni definite a tratti e per i sistemi
La sintassi di `cases` è simile a quella di `array`, ma senza
allineamento e con 2 sole colonne: le colonne sono separate da `&` e
ogni riga termina con `\\`.
La parentesi e l'allineamento sono automatici.

```
\[  
f(x)=\begin{cases}  
x^2 & \text{ per } x  
<1\\  
3 & \text{ per }  
1\leq x < 3\\  
\ln x & \text{ per }  
x\geq 3  
\end{cases} \]
```

$$f(x) = \begin{cases} x^2 & \text{per } x < 1 \\ 3 & \text{per } 1 \leq x < 3 \\ \ln x & \text{per } x \geq 3 \end{cases}$$

Equazioni numerate e no - 1

Abbiamo visto che un'espressione matematica può essere scritta

```
\[ espressione matematica \]
```

oppure (meno usata)

```
\begin{displaymath}  
  espressione matematica  
\end{displaymath}
```

Equazioni numerate e no - 2

È possibile anche usare la sintassi

```
\begin{equation}  
espressione matematica  
\label{etichetta}  
\end{equation}
```

che numera l'equazione.

Usando il pacchetto `amsmath` vi è anche la sintassi

```
\begin{equation*}  
espressione matematica  
\end{equation*}
```

che non numera l'equazione, ma gestisce meglio gli spazi.

Equazioni su più righe

Usando `amsmath` sono a disposizione ambienti per gestire equazioni che occupano più linee

- ▶ `gather gather*`
- ▶ `split`
- ▶ `multline multline*`
- ▶ `align align*`
- ▶ `flalign flalign*`

Teoremi e cose simili - 1

In matematica ci sono **Teoremi**, **Corollari**, **Definizioni** e molto altro.

Non ci sono ambienti predefiniti, ma c'è il comando `\newtheorem` (da scrivere nel preambolo) per avere ambienti in cui scrivere gli enunciati

```
\newtheorem{nome}{titolino}[contatore dominante]
\newtheorem{nome}[numerato come]{titolino}
```

- ▶ **nome** = nome dell'ambiente (teorema, lemma, ...). Se è teorema allora scriveremo `\begin{teorema}` enunciato `\end{teorema}`
- ▶ **titolino** = la parola che identifica l'enunciato; potrà essere Teorema, Lemma, **È quello che appare scritto**
- ▶ **contatore dominante** = il sezionamento che fa ricominciare la numerazione. Se è `chapter` a ogni capitolo ricomincia la numerazione
- ▶ **numerato come** = l'ambiente con cui si condivide la numerazione (deve essere già definito). Ad esempio `\newtheorem{lem}[teorema]{Lemma}` crea l'ambiente `lem` che avrà titolo "Lemma" e proseguirà la numerazione dei teoremi

Teoremi - Esempio 1

```
\newtheorem{thm}{Teorema}[section]
\newtheorem{cor}[thm]{Corollario}
\begin{thm}
\begin{equation}
    E=mc^2
\end{equation}
\end{thm}
\begin{cor}
    Un bel risultato
\end{cor}
```

Teorema 5.1.

$$E = mc^2 \quad (1)$$

Corollario 5.2. Un bel risultato

Teoremi - Esempio 2

```
\newtheorem{thm}{Teorema}[section]
\newtheorem{cor}{Corollario}[section]
\begin{thm}
    \begin{equation}
        E=mc^2
    \end{equation}
\end{thm}
\begin{cor}
    Un bel risultato
\end{cor}
```

Teorema 5.3.

$$E = mc^2 \quad (2)$$

Corollario 5.1. Un bel risultato

Teoremi e cose simili - 2

Scrivendo `\newtheorem*` gli enunciati non verranno numerati. `amsmath` e altri pacchetti come `ntheorem` permettono di definire in modo più fine la scrittura degli enunciati

Agli enunciati è possibile dare un nome

```
\begin{thm}[nome teorema]  
  testo  
\end{thm}
```

Teoremi - Esempio 3

```
\begin{thm}[di Euclide]
I numeri primi sono infiniti.
\end{thm}
```

Teorema 5.4 [di Euclide]. *I numeri primi sono infiniti.*

Dimostrazioni - 1

Il pacchetto `amsmath` ha anche l'ambiente `proof` per le dimostrazioni:

```
\begin{thm}[Sorpresa] Si  
ha che  
\[0=1\]  
\end{thm}  
\begin{proof}  
Da \((e^{2n\pi i})=1\)  
segue che  
\[e^{1+2n\pi i}=e\  
\[(e^{1+2n\pi i})^{1+2n\pi i}=e^{(1+2n\pi i)^2}=e\  
per cui  
\[e^{1+4n\pi i-4n^2\pi^2}=e\  
e infine  
\[e^{4n\pi i-4n^2\pi^2}=1\  
Mandando  $n\rightarrow+\infty$  si  
ha la tesi.  
\end{proof}
```

Teorema 5.5 [Sorpresa]. Si ha che

$$0 = 1$$

Dimostrazione. Da $e^{2n\pi i} = 1$ segue che

$$e^{1+2n\pi i} = e$$

$$(e^{1+2n\pi i})^{1+2n\pi i} = e^{(1+2n\pi i)^2} = e$$

per cui

$$e^{1+4n\pi i-4n^2\pi^2} = e$$

e infine

$$e^{4n\pi i-4n^2\pi^2} = 1$$

Mandando $n \rightarrow +\infty$ si ha la tesi. ■

Dimostrazioni - 2

Al termine della dimostrazione si usa inserire un simbolo

`\qedsymbol` che è solitamente un quadrato bianco.

Si può modificarlo con

`\renewcommand{\qedsymbol}{nuovo simbolo}`

Se necessario, col comando `\qedhere` si inserisce il simbolo. dove si desidera

il pacchetto `siunitx`

Per tutte le scienze in cui si ha a che fare con unità di misura è fondamentale l'utilizzo del pacchetto `siunitx`.

Il pacchetto usa come separatore decimale il punto, se si vuole la virgola bisogna caricarlo con

```
\usepackage[output-decimal-marker={,}]{siunitx}
```

Se le unità di misura sono poche si possono inserire “a mano” prestando attenzione agli spazi e alla simbologia corretta

siunitx - Esempi

```
\unit{kg.m.s^{-1}} \\  
\unit{\kilogram\metre\per\second} \\  
\unit[per-mode = symbol]  
\{\kilogram\metre\per\second} \\  

```

 kg m s^{-1} kg m s^{-1} kg m/s

```
\SI{23,4}{kg.m.s^{-2}} \\  
$r=\SI{0,8768(11)e-15}{m}$ \\  
\si{\joule\per\mole\per\kelvin} \\  
\si{j.mol^{-1}.K^{-1}} \\  

```

 $23,4 \text{ kg m s}^{-2}$ $r = 0,8768(11) \times 10^{-15} \text{ m}$ $\text{J mol}^{-1} \text{ K}^{-1}$ $\text{j mol}^{-1} \text{ K}^{-1}$

Ci sono pacchetti specifici per la fisica delle alte energie, per l'ottica (in particolare per grafici), per i diagrammi di Feynman, per i diagrammi elettrici, per la termodinamica.

Per quanto riguarda la chimica ci sono pacchetti che permettono tra l'altro di scrivere e disegnare facilmente le formule chimiche, gli schemi di reazioni, ecc.

Per la biologia ci sono pacchetti per gli alberi genealogici.

Le parti di un documento

Classe `article`

- ▶ “sezioni” (più correttamente in italiano “paragrafi”) `\section`
- ▶ “sottosezioni” `\subsection`
- ▶ “sottosottosezioni” `\subsubsection`
- ▶ “paragrafi” (in italiano più correttamente “capoversi”) `\paragraph`
- ▶ “sottoparagrafi” `\subparagraph`

Classi `report` e `book` hanno in più

- ▶ “parti” `\part`
- ▶ “capitoli” `\chapter`

di livello superiore rispetto a `section`

Sezionamento

La sintassi è (per tutti)

```
\section[opzioni]{titolo}
```

`opzioni` è il titolo che appare nella testatina della pagina e che viene stampato nell'indice.

Con la sintassi `\section*{titolo}` si produce il titolo, ma senza numero e senza inserirla nell'indice.

L'aspetto sia del titolo di un capitolo, una sezione, ..., che del titolo sulla testatina della pagina si modificano con il pacchetto `titlesec` che permette anche di modificare l'aspetto dell'indice.

Altri sezionamenti - 1

Nella classe `book` sono presenti dei “supersezionamenti” (le parti in italico sono necessarie)

- ▶ Materiale iniziale (`\frontmatter`): *Frontespizio*, Colophon, Dedicata, Sommario, *Indice generale*, Elenco delle figure, Elenco delle tabelle, Altri elenchi, Prefazione, Ringraziamenti, Introduzione non numerata. Non sono numerate, la numerazione delle pagine è con numeri romani minuscoli
- ▶ Materiale principale (`\mainmatter`): Introduzione numerata, *Capitoli*, una o più appendici numerate. Numera le sezioni e le pagine con numeri arabi (la numerazione della pagine riprende da 1)
- ▶ Materiale finale (`\backmatter`): una o più appendici non numerate, Glossario, *Bibliografia*, Indice analitico. Non sono numerate, la numerazione delle pagine prosegue quella principale

Altri sezionamenti - 1

Per produrre l'Indice Generale, l'Elenco delle Figure, l'Elenco delle Tabelle, Altri elenchi, la Bibliografia, il Glossario, l'Indice analitico esistono comandi appositi di $\text{\LaTeX}$ come

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

Nelle classi `article` e `report` esiste il “Sommario” (`abstract`) `\begin{abstract}...\end{abstract}` che contiene una breve introduzione. Nella classe `report` è, insieme al titolo, in una pagina a sé stante.

Titolo

```
\title{nome del documento}\\
\author{nomi degli autori}\\
\date{data}
.....
\begin{document}
    \maketitle
    .....
\end{document}
```

Se appare `\maketitle` il campo `\title` è obbligatorio.

Se ci sono più autori si separano con `\and` (vanno sulla stessa linea)
o con `\\` (vanno a capo)

Oggetti flottanti -1

Figure e Tabelle, richiedono un trattamento particolare, perché non possono essere spezzate e stampate su più pagine.

Sono uno degli argomenti più richiesti sui forum di $\text{\LaTeX}$ e costituiscono un'enorme fonte di frustrazione (e di cattiva composizione)

$\text{\LaTeX}$ rende mobili (o flottanti) le figure o le tabelle che non stanno sulla pagina corrente e le stampa eventualmente su una pagina successiva (e non dove l'utente vuole)

Comportamento tipograficamente corretto!

Se l'utente decide dove mettere una figura che la pagina contenente l'oggetto o quella precedente non sia “piena” di testo, ma presenti una parte bianca che è un effetto tipografico da evitare: una pagina tipograficamente ben formattata dovrebbe apparire da una certa distanza come un rettangolo uniformemente grigio.

Oggetti flottanti - 2

Figure

```
\begin{figure}[posizionamento]  
...  
\caption{testo della didascalia} \label{etichetta della figura}  
\end{figure}
```

Tabelle

```
\begin{table}[posizionamento]  
...  
\caption{testo della didascalia} \label{etichetta della tabella}  
\end{table}
```

Oggetti flottanti - 3

All'interno dell'ambiente `figure` o `table` si devono inserire la figura vera e propria o la tabella

`figure` e `table` sono “involucri” che consentono di creare riferimenti, inserire didascalie, ...

Il materiale presente in un ambiente `figure` o `table` viene trattato come un oggetto mobile e verrà inserito nel testo seguendo le indicazioni contenute nel parametro opzionale `posizionamento`.

Posizionamento di oggetti flottanti - 1

Carattere	Permesso di spostare l'oggetto...
h	qui (<i>here</i>), nel punto esatto in cui compare nel testo. Si usa principalmente per piccoli oggetti mobili.
t	in cima (<i>top</i>) a una pagina
b	in fondo (<i>bottom</i>) a una pagina
p	in una pagina speciale (<i>page</i>) contenente solo oggetti mobili.
!	senza considerare molti dei parametri interni (come il numero massimo di oggetti mobili che è possibile avere in una pagina) che potrebbero impedire a questo oggetto di essere spostato.

Se il posizionamento non viene specificato, le classi standard assumono che sia `[tbp]`.

Posizionamento di oggetti flottanti - 2

Per forzare $\text{\LaTeX}$ a inserire un oggetto flottante dove vogliamo il pacchetto `float` fornisce (tra le altre cose) l'opzione di posizionamento `H` (l'oggetto deve essere inserito esattamente nel punto indicato).

Se non si ha necessità di usare didascalie o riferimenti alla figura o alla tabella, il modo più semplice per posizionare figure e tabelle dove vogliamo è non usare gli ambienti `figure` o `table`, ma semplicemente `tabular` o `includegraphics`.

Posizionamento di oggetti flottanti - 3

L^AT_EX posiziona ogni oggetto mobile che incontra secondo le indicazioni dell'autore, se un oggetto non può essere posizionato nella pagina corrente, viene accumulato nella coda delle *figure* o delle *tabelle* (Code sono di tipo FIFO)

L^AT_EX mantiene strettamente l'ordine di apparizione originale per ogni tipo di oggetto mobile.

Quindi una figura che non può essere posizionata spinge tutte le figure successive alla fine del documento.

Per imporre a L^AT_EX di stampare immediatamente tutti gli oggetti mobili rimasti nelle code si usano i comandi

```
\clearpage  
\cleardoublepage
```

con il primo si inizia una nuova pagina, con il secondo una nuova pagina dispari.

Didascalia e riferimenti

Torniamo a

```
\begin{figure}[posizionamento]  
...  
\caption{testo della didascalia} \label{etichetta della figura}  
\end{figure}
```

Con il comando `\caption{testo}` è possibile specificare una didascalia per l'oggetto mobile. $\text{\LaTeX}$ aggiunge “Figura” o “Tabella” e un numero progressivo.

Il pacchetto `caption` permette di modificare in modo semplice l'apparenza delle didascalie.

Con `\label` e `\ref`, si possono creare riferimenti a oggetti mobili nel testo (come “Si veda la Figura 4 a pag. 23”).

Riferimenti incrociati - 1

Spesso in un testo leggiamo frasi del tipo “Come abbiamo visto nel Capitolo 3” oppure “Si veda la Figura 7 a pagina 45” o ancora “Dal Teorema 4 si deduce che “

Questo meccanismo si chiama **riferimenti incrociati** e ovviamente non è possibile gestirlo a mano!

Riferimenti incrociati - 2

In $\text{\LaTeX}$ il funzionamento è semplicissimo

- ▶ si scrive `\label{etichetta}` subito dopo l'oggetto a cui si dovrà fare riferimento (figura, tabella, sezionamento, teorema, equazione, ...): `etichetta` deve essere unica
- ▶ si scrive `\ref{etichetta}` per fare riferimento all'oggetto
- ▶ si scrive `\pageref{etichetta}` per fare riferimento alla pagina in cui è oggetto

È **estremamente** utile che `etichetta` contenga sia della forma abbreviazione:parola chiave come

```
\label{fig:la gioconda} o  
{sec:riferimenti incrociati}
```

Gli oggetti che permettono i riferimenti incrociati sono i sezionamenti, le tabelle, le figure, gli elenchi, le note, i teoremi, le equazioni, ...

In *Overleaf* e in molti editor scrivendo `\ref` appare un elenco delle `label` definite.

Bibliografia

- ▶ utilizzando l'ambiente `thebibliography`;
- ▶ utilizzando il programma `BibTeX`;
- ▶ utilizzando il programma `biber`.

Vedremo solo il primo metodo perché gli altri prevedono la creazione di un database bibliografico esterno

L'ambiente `thebibliography`

```
\begin{thebibliography}{etichetta più lunga}
    \bibitem[etichetta]{chiave} voce bibliografica
    ...
\end{thebibliography}
```

- ▶ `stringa` è una sequenza di cifre, che serve al programma per sapere quanto possono essere lunghe le etichette con cui $\text{\LaTeX}$ identifica la referenza
- ▶ `chiave` ha lo stesso scopo di `label` nei riferimenti incrociati
- ▶ `etichetta` è un'etichetta personalizzata che può comparire nella bibliografia e nelle citazioni invece di quella predefinita

Le citazioni bibliografiche

Le voci bibliografiche vengono richiamate con il comando

Le voci bibliografiche vengono richiamate con il comando

`\cite[testo]{chiave}` oppure `\cite{lista di chiavi}`

- ▶ `chiave` funziona come `ref` per i riferimenti incrociati; è possibile citare più voci bibliografiche separandole con virgole
- ▶ `testo` può essere il riferimento a una pagina o a un capitolo specifico dell'opera

Bibliografia - Esempio

```
\begin{thebibliography}{999}
\bibitem{MR957792} Bárány I. {On the minimal ring containing the boundary of a
convex body.} Acta Sci. Math. (Szeged). \textbf{52}, 93-100 (1988)

\bibitem{Bokowski1986} Bokowski J., Heil E. { Integral representations of
quermassintegrals and Bonnesen-style inequalities.} Arch. Math \textbf{47}, 79-89
(1986)

\bibitem{Bonnesen1929} Bonnesen T. {Les problèmes des isopérimètres et des
isépiphanes.} Gauthier-Villars, Paris. (1929)

\bibitem{Bottema1933} Bottema O. {Eine obere Grenze für das isoperimetrische Defizit
ebener Kurven.} Ned. Akad. Wet. Proc. \textbf{66} pp. 442-446 (1933)

\bibitem{Chouikha} Chouikha A. {Bonnesen-type inequalities and applications.}
Generalized Convexity, Generalized Monotonicity And Applications. \textbf{77} pp.
173-181 (2005)

\bibitem{MR3564432} Cufí J., Reventós A. {A lower bound for the isoperimetric
deficit.} Elem. Math. \textbf{71}, 156-167 (2016)

\bibitem{Favard1929} Favard J. {Problèmes d'extremums relatifs aux courbes convexes
(premier mémoire).} Annales scientifiques de l'École Normale Supérieure, Série 3
(1929)
\end{thebibliography}
```

Le citazioni saranno della forma

```
\cite[pag. 34]{Bonnesen1929} oppure \cite{MR3564432} o
\cite{MR3564432, Favard1929}
```

Esistono molti pacchetti che permettono una gestione più flessibile della bibliografia.

È molto usato il pacchetto `natbib` che permette di scegliere facilmente sia la “forma” della citazione (tra parentesi tonde, quadre, graffe, come nota) sia lo “stile” (numerico, autore-anno, ecc.)

L'ambiente `thebibliography` ha alcuni inconvenienti:

- ▶ obbliga l'autore a scrivere ogni volta da capo la bibliografia per ogni documento, anche se c'è solo qualche opera in più o se deve modificare lo stile;
- ▶ in caso di aggiornamento di un documento, bisogna modificare le bibliografie di tutti i documenti in cui è citato.

Il problema si supera usando un database delle opere, e di generare automaticamente la bibliografia a partire da quest'ultimo.

L'indice generale - 1

L'indice viene creato con il comando `\tableofcontents`.

Riporta i capitoli, i paragrafi, . . . , fino ai sottocapoversi

È possibile evitare l'aggiunta di alcuni elementi all'indice usando la forma con asterisco del comando `chapter`, `section`, ecc.

È possibile aggiungere un campo all'indice con il comando `\addcontentsline{toc}{livello}{campo}` che aggiunge il campo al livello indicato (capitolo, paragrafo, . . .).

La bibliografia non è inserita nell'indice

```
\addcontentsline{toc}{chapter}{Bibliografia}
```

inserisce la bibliografia nell'indice come un capitolo.

L'indice generale - 2

Se un titolo è troppo lungo per stare “comodamente” nell'indice (o nella testatina se esiste) lo si può sostituire con un titolo alternativo più breve, da inserire nell'argomento facoltativo degli stessi comandi:

```
\chapter[Questo titolo è breve]{Questo è un titolo lunghissimo  
e non posso metterlo nell'indice perché non è elegante avere un  
titolo che va a capo}
```

Altri indici

Possiamo avere anche

- ▶ la *lista delle figure* (lof) inserita con il comando
`\listoffigures`
- ▶ la *lista delle tabelle* (lot) inserita con il comando
`\listoftables.`

Inoltre si possono usare altre liste come la *lista dei simboli* usando pacchetti appositi

Il pacchetto `beamer` - 1

Il pacchetto `beamer` è un ottimo strumento per effettuare presentazioni sia di tipo didattico che scientifico ed ha a disposizione un notevole numero di “effetti” video come transizioni e box.

L'autore del pacchetto raccomanda, per ottenere una presentazione efficace, di non eccedere con sfondi vistosi ed effetti video, che tendono a distrarre l'uditorio

Il pacchetto `beamer` - 2

L'output di `beamer` è un file pdf e quindi è visualizzabile su qualsiasi device su cui sia installato un lettore di file pdf

Però non con tutti i visualizzatori di file pdf e su tutte le piattaforme è possibile vedere gli effetti che vengono creati come ad esempio la trasparenza. Inoltre è indispensabile impostare la visualizzazione a pieno schermo.

Il pacchetto `beamer` - 3

Le slide ottenute possono contenere anche i “pulsanti” per muoversi nella presentazione ed è possibile avere una barra contenente un indice generale della presentazione sempre visibile

Poiché `beamer` è una classe il suo uso per un utente $\text{\LaTeX}$ non si discosta molto da quello delle altre classi e, come sempre in $\text{\LaTeX}$, permette di concentrarsi sul contenuto più che sull’aspetto esteriore del documento.

Il pacchetto beamer - Il preambolo 1

```
\documentclass{beamer}
\mode<presentation>
{\usetheme{Boadilla} \usecolortheme{albatross}}
\usefonttheme{structuresmallcapsserif}
\usepackage[italian]{babel}
% o altra lingua
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\title[Titolo Abbreviato] {Titolo completo}
\date{15 agosto 2024}
\author {Autore 1\inst{1} \and Autore 2\inst{2}}
\institute[Università Cattolica di Milano]{
\inst{1}
Dipartimento di Matematica\\
Università Cattolica di Milano
\and
\inst{2}%
Dipartimento di Economia\\
Università Cattolica di Milano}
\pgfdeclareimage[height=0.5cm]{logo}{nome del file del logo}
\logo{\pgfuseimage{logo}}
```

Il pacchetto `beamer` - Il preambolo 1

Nel preambolo è già possibile vedere alcune caratteristiche di beamer:
i **modi** e i **temi**

Con un file beamer si possono ottenere

- ▶ la presentazione,
- ▶ un file contenente le slides pronte per la stampa (quindi senza effetti di transizione, forse meno colorate, spesso con più slides per pagina)
- ▶ un file contenente una versione a stampa della presentazione sotto forma di articolo.

Il file beamer può contenere istruzioni e testo specifici per ognuna o per più o tutte queste modalità.

Il pacchetto beamer - Il preambolo 2

Le istruzioni

`\usetheme{Boadilla}` e

`\usecolortheme{albatross}`

sono argomento del comando

`\mode<presentation>`

e si riferiscono solo alla modalità presentazione, mentre l'istruzione

`\usefonttheme{structuresmallcapserif}`

si riferisce a tutte le modalità possibili.

Ci sono temi globali che definiscono tutti gli elementi della presentazione (colori, caratteri, la presenza o meno di una barra di navigazione) e temi che modificano solo alcuni aspetti

`\usetheme{Boadilla}` è del primo tipo,

`\usecolortheme{albatross}`, del secondo

Il corpo del documento 1

La classe `beamer` non ha i capitoli, ma le parti (`part`), le sezioni e le sottosezioni (`section` e `subsection`). Questa suddivisione viene usata nell'indice e (se esiste) nella barra di navigazione.

L'analogo della pagina è il **frame** il cui contenuto può essere visualizzato in un'unica soluzione o in tempi successivi, eventualmente con effetti.

Il frame può essere introdotto da un **effetto di transizione** che può essere parzialmente modificato.

Il corpo del documento 2

Il frame è un ambiente

```
\begin{frame}  
    \transglitter  
    testo molto interessante  
\end{frame}
```

Il comando `\titlepage` crea un frame con titolo, autore, ecc.,

All'interno di un frame il comando

`\frametitle{titolo del frame}` crea un titolo evidenziato
per quel frame

Il contenuto di un frame dovrebbe risultare in un'unica pagina a video,
l'opzione

`[allowframebreaks]`

permette al contenuto di espandersi su più pagine (da evitare!)

Il corpo del documento - Slides 1

Il contenuto di un frame può essere visualizzato a passi: ognuno di questi successivi passi viene detto *slide*

Il contenuto e il tempo di visualizzazione di ogni slide può essere modificato

- ▶ `\pause`: beamer mostra il contenuto del frame fino al comando e aspetta un input per avanzare
- ▶ `\onslide<...>{testo}` mostra il testo contenuto tra parentesi graffe nelle slides specificate tra `<>`: è possibile indicare una sola slide `<3>`, più slides `<3, 5, 9>`, un intervallo `<2-4>`, tutte le slides fino a (a partire da) una specificata `<-4>` (`<2->`).

Il corpo del documento - Slides 2

Altri comandi simili a `\onslide`

- ▶ `\only`
- ▶ `\uncover`
- ▶ `\visible`
- ▶ `\alt`

Beamer può trattare il testo non visibile in due modi diversi:
come testo invisibile (che occupa spazio e può eventualmente avere effetti di trasparenza)
come testo che viene inserito solo nelle slides specificate e viene “gettato via” nelle altre

Il comando `\only` è del secondo tipo
il comando `\uncover` è del primo tipo
il comando `\onslide` di default è del primo tipo, ma può assumere dei modificatori per ottenere risultati diversi.

Il corpo del documento - Slides 3

Molti comandi $\text{\LaTeX}$ e ambienti (come `theorem`, i comandi di formattazione del testo, ...) accettano come opzione `<...>`:

```
\begin{frame}
  \textbf{Linea in grassetto su tutte e tre le slides.}\\
  \textbf<2>{Linea in grassetto solo sulla seconda
slide.}\\
  \textbf<3>{Linea in grassetto solo sulla terza slide.}\\
\end{frame}
```

Linea in grassetto su tutte e tre le slides.

Linea in grassetto solo sulla seconda slide.

Linea in grassetto solo sulla terza slide.

Elenchi passo-passo

Un caso importante di visualizzazione passo-passo sono gli elenchi ed esiste un'opzione da dare a `itemize` o `enumerate`: `<+>`:

```
\begin{itemize}[<+>]
  \item Questo appare dalla prima slide.
  \item questo appare dalla seconda slide .
  \item<1-> Anche questo appare dalla prima slide essendo
specificato nell'item
  \item questo appare dalla terza slide
\end{itemize}
```

- ▶ Questo appare dalla prima slide.
- ▶ questo appare dalla seconda slide .
- ▶ Anche questo appare dalla prima slide essendo specificato nell'item
- ▶ questo appare dalla terza slide

Evidenziare testo

La classe beamer ha a disposizione comandi ed ambienti per evidenziare parti di testo.

I due comandi principali sono

`\alert<...>{testo}` e `\structure<...>{testo}`

che modificano grandezza e/o colore del testo e possono essere usati come opzione negli elenchi

```
\begin{itemize}[<+ -| alert@+>]
  \item uno
  \item due
  \item tre
\end{itemize}
```

In questo esempio quando un item appare, viene evidenziato in blu.

- ▶ uno
- ▶ due
- ▶ tre

Box 1

Un altro modo per sottolineare alcuni concetti della presentazione è usare dei box colorati. Alcuni ambienti come `theorem`, `definition`, `example` costruiscono attorno al testo una “scatola” colorata

Box 2

Si possono creare box con i comandi:

```
\begin{block}  
    {Titolo}  
    testo  
\end{block}  
\begin{alertblock}  
    {Titolo}  
    testo  
\end{alertblock}
```

Titolo

testo

Titolo

testo

Box 2

È possibile creare box con altri colori con gli ambienti

`beamercolorbox` e `beamerboxesrounded`

```
\setbeamercolor{postit}{fg=black,bg=yellow}
\begin{beamercolorbox}[sep=1em,wd=8cm]{postit}
  Appiccicami da qualche parte!
\end{beamercolorbox}

\setbeamercolor{uppercol}{fg=white,bg=orange}
\setbeamercolor{lowercol}{fg=black,bg=green}
\begin{beamerboxesrounded}[upper=uppercol,lower=lowercol,shadow=true]{Teorema}
  $A = B$.
\end{beamerboxesrounded}
```

Appiccicami da qualche parte!

Theorem

$A = B.$

Tabelle, figure, suoni ed animazioni

Tutti gli effetti possono essere applicati anche a figure e tabelle, per cui è possibile far apparire una riga alla volta di una tabella o solo alcune parti di una figura.

Beamer permette anche di ingrandire parti di una figura (effetto zoom) con il comando `\framezoom`.

È possibile inserire all'interno della presentazione file multimediali sia audio che video (i file stessi non sono inseriti nel file pdf che contiene solo un collegamento: quindi è necessario avere i file multimediali insieme al file della presentazione): a seconda della configurazione del computer e del visualizzatore sarà possibile mostrare il file all'interno della presentazione o con un visualizzatore esterno.

PSTricks

PSTricks è un collezione di macro di interfaccia tra $\text{T}_{\text{E}}\text{X}$ (e $\text{\LaTeX}$) e il linguaggio PostScript: è un pacchetto molto grosso e ha centinaia di estensioni

PSTricks consente di disegnare tutti i tipi di oggetti, grafici 2D e 3D, a colori e non, e molto altro.

Usando PSTricks, un qualsiasi disegno viene “tracciato” attraverso una serie di comandi.

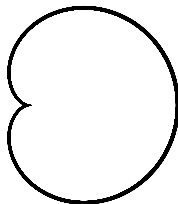
PSTricks - Esempi

Circonferenza con centro nell'origine degli assi e raggio 3

```
\pscircle(0,0){3}
```

Cardioide di equazione in coordinate polari, $\rho(\theta) = 1 + \cos(\theta)$

```
\parametricplot[algebraic,plotpoints=1500]  
{0}{360}{0.5+cos(t)+0.5*cos(2*t)|sin(t)+0.5*sin(2*t)}
```



pstricks-add

Il pacchetto `pstricks-add` estende le potenzialità di `PSTricks` e ne facilita l'uso in molti ambiti.

Ad esempio, consente di sovrapporre colori e immagini, permette di tracciare in modo più immediato i grafici delle funzioni e di scriverne l'espressione in notazione algebrica (quella a cui siamo abituati) anziché in notazione inversa polacca (quella standard di `PSTricks`), e molto altro ancora.

L'ambiente pspicture

Gli oggetti grafici creati con PSTricks devono essere contenuti all'interno dell'ambiente *pspicture*.

```
\begin{pspicture}(a, b)(c, d)
    ...
\end{pspicture}
```

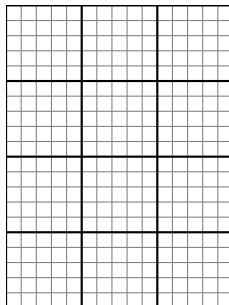
Il comando crea una “scatola” in un piano cartesiano immaginario con vertice inferiore sinistro in (a, b) e vertice superiore destro in (c, d) : gli oggetti verranno sistemati nella scatola partendo da tali coordinate. L'oggetto può fuoriuscire dalla scatola.

Se si vuole tagliare ciò che esce dalla scatola si usa `pspicture*`

La griglia

Il comando `\psgrid` crea un rettangolo con un reticolato, che riempie l'ambiente `pspicture`

```
\begin{pspicture} (0,0) (3,4)  
\psgrid  
\end{pspicture}
```



I punti

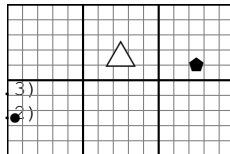
Il comando

`\psdots[opzioni] (x1,y1) (x2,y2) . . . (xn,yn)` disegna un punto per ognuna delle coordinate specificate.

- ▶ *dotstyle=stile*: consente di modificare l'aspetto dei punti (triangoli, cerchi, quadrati, ecc.);
- ▶ *dotsize=misura*: consente di specificare la grandezza assoluta dei punti;

I punti - esempi

```
\begin{pspicture}(0,0)(3,2)
\psgrid
\psdots (0.1,0.5)
\psdots[dotstyle=triangle,dotsize=10pt](1.5,1.3)
\psdots[dotstyle=pentagon*,dotsize=5pt](2.5,1.2)
\end{pspicture}
```



Le linee - I segmenti

Per i segmenti si usa il comando:

```
\psline[opzioni] (coordinate punto iniziale) (coordinate punto  
finale)
```

```
\begin{pspicture} (0,0) (2,1)  
\psline[linewidth=2pt]{<->}(0,2) (2,2)  
\psline[linestyle=dashed,linecolor=blue,linewidth=2pt]%(  
(0,1) (2,1)  
\psline[linestyle=dotted] (0,0) (2,0)  
\psline (0,-1) (2,-1)  
\end{pspicture}
```

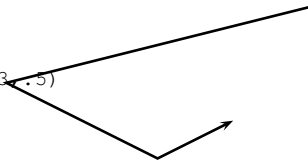


Si possono aggiungere frecce: alla linea si aggiungerà > o < tra parentesi graffe

Le linee - Le spezzate

Le spezzate hanno lo stesso comando, solo che la linea viene fatta passare per più di due punti:

```
\begin{pspicture} (0,0) (4,2)  
\psline[linewidth=1pt]{->} (4,2) (0,1) (2,0) (3,1.5)  
\end{pspicture}
```



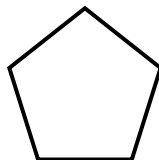
Le linee - I poligoni

Per i poligoni si usa il comando

```
\pspolygon[opzioni] (x0,y0) (x1,y1) (x2,y2) ... (xn,yn),
```

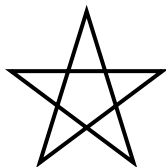
La spezzata si chiude automaticamente

```
\begin{pspicture} (-0.3,-0.3) (2.3,2.3)  
\pspolygon[linewidth=1.5pt]%  
(0,1.21) (0.38,0) (1.62,0) (2,1.21) (1,2)  
\end{pspicture}
```



Le linee - I poligoni

```
\begin{pspicture}(-0.3,-0.3)(2.3,2.3)
\pspolygon[linewidth=1.5pt]%
      (0,1.21)(2,1.21)(0.38,0)(1,2)(1.62,0)
\end{pspicture}
```



Stili delle linee

Si possono cambiare lo stile, lo spessore e il colore delle linee:

1. **L'opzione *linestyle*:** permette di cambiare lo stile delle linee (che di default sono continue). Alcuni degli stili disponibili , sono
 - ▶ la linea tratteggiata (*linestyle=dashed*),
 - ▶ la linea a puntini (*linestyle=dotted*)
 - ▶ la linea "invisibile" (*linestyle=none*)
2. **L'opzione *linicolor*:** consente di cambiare il colore di una linea (che di default è nero).
3. **L'opzione *linewidth*:** consente di gestire lo spessore della linea che stiamo disegnando, che di default è di 0.8 punti.

Riempimento

1. **L'opzione *fillcolor*:** serve per determinare il colore con cui viene riempito l'interno di una figura.

Il comando funziona solo se viene anche specificato uno stile di riempimento, mediante l'opzione *fillstyle*

2. **L'opzione *fillstyle*:** determina lo stile con il quale un oggetto viene riempito.

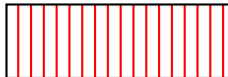
- ▶ *none*: è lo stile di default la figura non viene riempita;
- ▶ *solid*: l'oggetto viene riempito in modo uniforme nel colore determinato da `fillcolor` (di default bianco);
- ▶ *vlines*, *vlines**, *hlines*, *hlines**, *crosshatch*, *crosshatch**: riempie la figura di linee rispettivamente verticali, orizzontali o incrociate;
- ▶ *transparent*: consente di riempire l'interno della figura di colore "trasparente" (è possibile scegliere il grado di trasparenza)

Riempimento - esempi

```
\begin{pspicture}(3,3)
\psframe[fillstyle=solid,fillcolor=red](0,1)(3,2)
\end{pspicture}
```



```
\begin{pspicture}(3,3)
\psframe[fillstyle=vlines,hatchcolor=red,hatchangle=0]%(0,1)(3,2)
\end{pspicture}
```



Le forme geometriche di base

In PSTricks si creano facilmente le forme geometriche di base: rettangoli, cerchi, ellissi, . . .

Tutti i comandi hanno due versioni, una normale e una con asterisco che riempie l'oggetto creato dello stesso colore del bordo.

Rettangoli

Sono creati dal comando `\psframe`.

```
\begin{pspicture} (0.5,1.8) (3.5,3.3) )  
\psframe[linewidth=2pt] (0.7,2) (3.3,3)  
\end{pspicture}
```



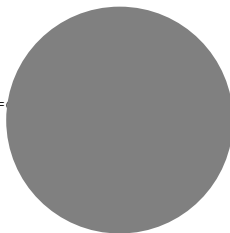
Cerchi

Si ottengono con il comando

```
\pscircle[opzioni] (x0,y0) {raggio}
```

dove (x_0, y_0) sono le coordinate del centro e *raggio* è la misura del raggio.

```
\begin{pspicture} (1,-1) (5,3)  
\psset{linewidth=1pt,linecolor=  
\pscircle* (3,1) {1.5}  
\end{pspicture}
```



Parabole

Si ottengono mediante il comando

```
\parabola[opzioni] (x0,y0) (x1,y1)
```

che disegna una parabola con asse parallelo all'asse delle ordinate, passante per il punto di coordinate (x_0,y_0) e avente vertice nel punto di coordinate (x_1,y_1) .

```
\begin{pspicture} (3,3)  
\parabola[linestyle=dashed] (0,1) (1,2.5)  
\end{pspicture}
```



Grafici di funzioni

Per disegnare i grafici delle funzioni è utile usare il pacchetto `pstricks-add` che permette di usare la notazione solita per le operazioni anziché la notazione inversa polacca (RPN)

Gli assi si disegnano con:

`\psaxes[opzioni]{frecce}(x0,y0)(x1,y1)(x2,y2)`,
dove (x_1,y_1) e (x_2,y_2) sono gli angoli opposti del rettangolo che delimita gli assi; (x_0,y_0) sono le coordinate del punto di intersezione degli assi (l'origine).

È possibile modificare l'unità di misura di uno o entrambi gli assi e anche usare la scala logaritmica.

Funzioni 2D

Il comando che ci consente di disegnare i grafici delle funzioni è:

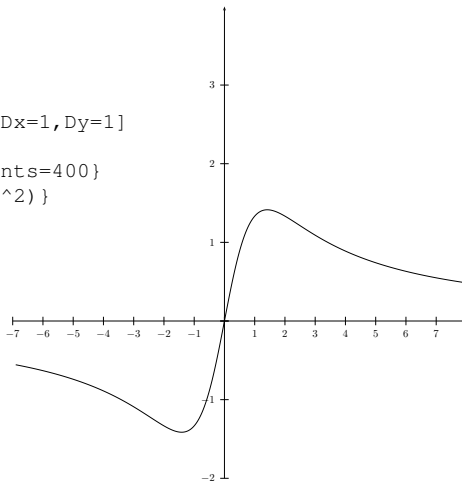
```
\psplot{ascissa iniziale}{ascissa finale}{funzione}
```

Vogliamo disegnare il grafico della funzione:

$$y = \frac{4x}{x^2 + 1}$$

Funzioni 2D - esempio

```
\psset{yunit=2.6cm,xunit=1cm}  
\begin{pspicture}(-7,-2)(8,4)  
\psaxes[labels=all,ticks=all,Dx=1,Dy=1]  
{->} (0,0) (-7,-2) (8,4)  
\psset{algebraic=true,plotpoints=400}  
\psplot{-6.9}{7.9}{(4*x)/(2+x^2)}  
\end{pspicture}
```



I nodi

Un *nodo* è una forma che può contenere un testo al proprio interno: le funzioni sono di aggiungere testo al disegno e di identificare un punto. Si usa il pacchetto `pst-node`.

- ▶ `\rnode[punto di riferimento]{nome}{testo}`: crea un nodo etichettato con *nome*, per il quale appare quanto scritto in *testo*; i connettori puntano al punto di riferimento indicato nella prima parentesi quadra;
- ▶ `\pnode(x,y){nome}`: crea un nodo nel punto di coordinate (x,y) senza occupare spazio;
- ▶ `\cnode[opzioni](x,y){raggio}{nome}`: crea un nodo circolare, di raggio fissato dal valore nella prima parentesi graffa, nel punto di coordinate (x,y) (la versione asteriscata crea un cerchio pieno, quella senza asterisco vuoto).

Collegare i nodi

I nodi possono essere collegati tra loro da linee di vario tipo.

Il comando

```
\ncline[opzioni]{frecce}{nodo1}{nodo2}
```

permette di unire due nodi con una linea richiamando i nodi da unire con le loro etichette.

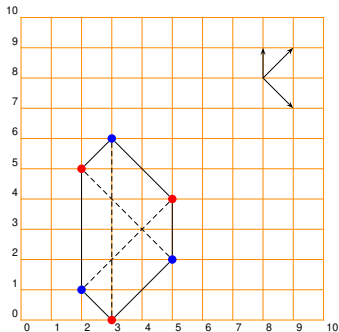
I nodi - esempio

```
\begin{pspicture}(0,0)(10,10)
\psgrid[subgriddiv=1,gridcolor=DarkOrange]
\cnode*[linecolor=red](3,0){4pt}{A}
\cnode*[linecolor=red](5,4){4pt}{B}
\cnode*[linecolor=red](2,5){4pt}{C}
\cnode*[linecolor=blue](2,1){4pt}{D}
\cnode*[linecolor=blue](5,2){4pt}{E}
\cnode*[linecolor=blue](3,6){4pt}{F}
\ncline{A}{D} \ncline{C}{D}
\ncline{A}{E} \ncline{C}{F}
\ncline{E}{B} \ncline{B}{F}
\ncline[linestyle=dashed]{B}{D}
\ncline[linestyle=dashed]{C}{E}
\ncline[linestyle=dashed]{A}{F}
\psline[arrowsize=5pt]{->}(8,8)(8,9)
\psline[arrowsize=5pt]{->}(8,8)(9,9)
\psline[arrowsize=5pt]{->}(8,8)(9,7)
\end{pspicture}
\vspace{.5cm}
\Huge{\[\left\{0,4,5\right\}

$$\overset{2}{=} \left\{1,2,6\right\}$$


$$\left.\right\}$$

```



Altri effetti

Con PSTricks è possibile lavorare sul testo, trasformando scritte (e anche espressioni matematiche) in diversi modi.

Esistono inoltre numerosi altri pacchetti per la grafica 3d, per i circuiti, per i frattali, ecc.

TikZ e PGF - 1

Il pacchetto `PGF` e la sua interfaccia (non grafica) `TikZ` permettono di costruire figure mediante comandi come `PSTricks`, ma l'output può essere sia un file PostScript, sia un file PDF, senza l'intervento di programmi esterni, ma con l'usuale compilazione con $\text{\LaTeX}$ o $\text{PDF\LaTeX}$.

La compatibilità con $\text{PDF\LaTeX}$ ha come conseguenza una minore potenza del pacchetto rispetto a `PSTricks` perché il linguaggio PDF è meno potente del linguaggio PostScript.

TikZ e PGF - 2

Tutte le operazioni grafiche vengono eseguite dalla libreria `pgf` e `TikZ` è solo un'interfaccia che possiede comandi semplificati per l'uso della libreria

Esistono moltissime librerie specializzate di `pgf` ad esempio per disegnare diversi tipi di frecce o per usare sfondi, disegnare buffi animaletti, ecc.

tikzpicture

I comandi vengono inseriti nell'ambiente `tikzpicture`.

Il comando fondamentale è

`\path`

che descrive un oggetto (segmento, cerchio, rettangolo, ecc.).

Quando a questo comando, in un qualsiasi punto, viene data l'opzione:

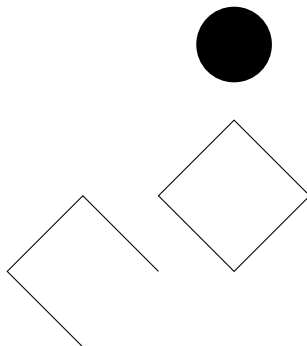
- ▶ `[draw]`, l'oggetto viene disegnato;
- ▶ `[fill]`, l'oggetto viene riempito;
- ▶ `[fill, draw]`, l'oggetto viene disegnato e riempito.

È possibile scrivere direttamente

- ▶ `\draw,`
- ▶ `\fill,`
- ▶ `\filldraw`

Esempio

```
\begin{tikzpicture}
\path(1,0)--(0,1)--(-1,0)--(0,-1)
[draw];
\draw(3,1)--(2,2)--(1,1)--(2,0)--cycle;
\fill(2,3) circle (5mm);
\end{tikzpicture}
```



Segmenti

Il comando

```
\draw (x1,y1) -- (x2,y2)
```

disegna un segmento (una spezzata se si indicano più di due punti),

```
(x1,y1) (x2,y2)
```

indica che la “penna” deve spostarsi dal punto $(x1,y1)$ al punto $(x2,y2)$ senza tracciare nulla.

Segmenti - esempio

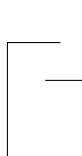
```
\begin{tikzpicture}  
\draw (0,0)--(1,0)  
(0,-1)--(1,-1)--(-.5,-.5);  
\end{tikzpicture}
```



Snakes

Due punti possono essere uniti non solo con un segmento, ma anche con una coppia di segmenti uno orizzontale e uno verticale (o viceversa) o mediante linee “bufe”, ad esempio a zigzag, a molla, ecc., (l'autore chiama tali tipi di linea *snakes*); per disegnare tali linee è necessario nel preambolo caricare la libreria apposita con il comando `\usepackage{pgflibrarysnakes}`.

```
\begin{tikzpicture}
\draw (0.5,0)-|(1,1) (0,-1)|-(.7,.5);
\draw [snake=coil,segment length=4pt]
(0,1.5)--(2,1.5);
\end{tikzpicture}
```



Poligonalì

Aggiungendo `--cycle` al termine dei punti di una spezzata, si ottiene una poligonale

```
\begin{tikzpicture} [line width=10pt]
\draw(0,0)--(1,1)--(1,0)--(0,0);
\draw(2,0)--(3,1)--(3,0)-- cycle;
\end{tikzpicture}
```



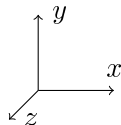
3d

In TikZ è facile disegnare in un ambiente 3d: in effetti è sufficiente usare tre coordinate per farlo o specificare che si lavora in un sistema di coordinate xyz .

Le coordinate x e y sono come quelle del piano, mentre il punto $(0, 0, 1)$ viene rappresentato assonometricamente con il punto di coordinate $\left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$.

3d - esempio

```
\begin{tikzpicture}[->]
\draw (0,0,0)--(1,0,0)
node[above=2pt]{$x$};
\draw (0,0,0)--(xyz cs:y=1)
node[right=2pt]{$y$};
\draw (0,0,0)--(0,0,1)
node[right=2pt]{$z$};
\end{tikzpicture}
```



Rettangoli, cerchi, ellissi, parabole - 1

Rettangoli, cerchi, ellissi, parabole hanno comandi specifici per essere disegnati.

- ▶ Per disegnare un rettangolo con vertici opposti in (x_0, y_0) e (x_1, y_1)

```
\path (x0,y0) rectangle (x1,y1);
```

- ▶ Per disegnare un cerchio

```
\path (x0,y0) circle (raggio);
```

dove (x_0, y_0) è il centro (se non è specificato il centro è il punto corrente) Alla fine del comando il punto corrente è il centro.

Rettangoli, cerchi, ellissi, parabole - 2

- Per disegnare un ellisse

```
\path (x0,y0) ellipse (a and b);
```

dove $(x0,y0)$ è il centro, a il semiasse nella direzione dell'asse x e b il semiasse nella direzione dell'asse y .

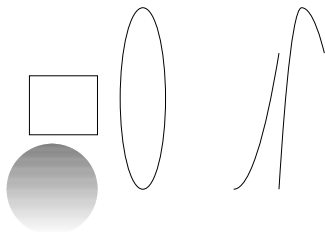
- Per disegnare una parabola abbiamo i seguenti comandi

```
\path (x0,y0) parabola (x1,y1);  
\path (x0,y0) parabola bend (x2,y2) (x1,y1);
```

Entrambi i comandi tracciano un arco di parabola passante per il punto corrente $(x0,y0)$ (che può non essere indicato) e il punto $(x1,y1)$, ma nel secondo caso si impone che il vertice sia in $(x2,y2)$ mentre nel primo il vertice è nel punto corrente.

Rettangoli, cerchi, ellissi, parabole - esempi

```
\begin{tikzpicture}
\draw (-.5,1.2) rectangle
(1,2.5);
\shade (0,0) circle (1 cm);
\draw (2,2) ellipse (.5 and 2);
\draw (4,0) parabola (5,3);
\draw (5,0) parabola bend
(5.5,4) (6,3);
\end{tikzpicture}
```



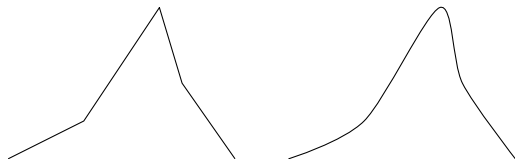
Grafici per punti

```
\draw plot[opzioni]{coordinates{(x0,y0) (x1,y1) ... (xn,yn)}};
```

disegna una curva che unisce i punti indicati: la curva è una spezzata, ma con l'opzione `smooth` viene “lisciata”.

Si possono leggere le coordinate da un file esterno

```
\draw plot coordinates{(0,0) (1,.5) (2,2) (2.3,1) (3,0)};  
\draw[smooth] plot coordinates{(0,0) (1,.5) (2,2) (2.3,1) (3,0)};
```



Grafici di funzioni - 1

```
\draw plot[opzioni]{espressione con coordinate}
```

dove l'espressione con coordinate contiene una macro speciale la variabile, che di default è `\x`; l'espressione viene valutata per diversi valori della variabile e la tabella di coordinate risultante viene disegnata.

Grafici di funzioni - 2

- ▶ `smooth` Unisce i punti del grafico “lisciando” la curva
- ▶ `variable= macro (la variabile),`
- ▶ `samples= numero di punti` (il numero di punti in cui l'espressione viene valutata, di default 25),
- ▶ `domain= inizio:fine` (il dominio in cui i punti vengono presi, di default -5:5),
- ▶ `\samples at={lista di numeri}` (fornisce esplicitamente in quali punti deve essere valutata l'espressione, tale opzione annulla eventuali dichiarazioni di `samples` o `\domain`).

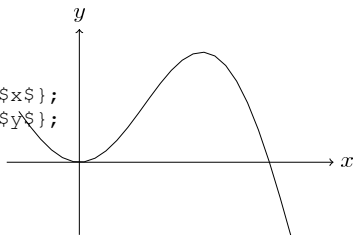
Ci sono poi le solite opzioni su spessore della linea, colore, stile, ...

Grafici di funzioni - 3

La simbologia matematica è quella usuale, le funzioni seno e coseno sono calcolate in gradi, per calcolarle in radianti si deve usare esplicitamente l'unità r , π si indica con `pi` mentre "e" deve essere calcolato con `exp ( 1 )` ; se si usano parentesi (solo tonde!) è necessario racchiudere l'espressione tra parentesi graffe.

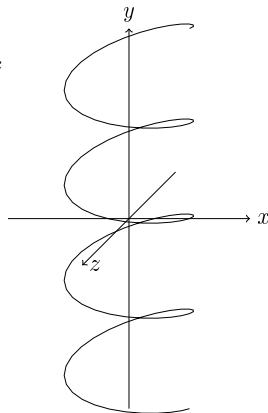
Grafici di funzioni - esempio 1

```
\begin{tikzpicture}[b][domain=-1:3.5]
\draw[>-] (-1.2,0)--(4.2,0)node[right]{$x$};
\draw[>-] (0,-1.2)--(0,4.2)node[above]{$y$};
\draw plot (\x,{\x*sin(\x r)});
\end{tikzpicture}
```



Grafici di funzioni - esempio 2

```
\begin{tikzpicture}
\draw[>-] (-2,0,0)--(2,0,0)node[right]{$x$};
\draw[>-] (0,-\pi,0)--(0,\pi,0)node[above]{$y$};
\draw[>-] (0,0,-2)--(0,0,2)node[right]{$z$};
\draw plot
[domain=-\pi:\pi,variable=\t,samples=100]
({cos(4*\t r)},\t,{sin(4*\t r)});
\end{tikzpicture}
```



Nodi - 1

Il comando

```
node[opzioni] {testo}
```

permette di inserire testo.

In realtà permette di dare a un punto un nome per referenze future e inserire del testo in una figura (cerchio o rettangolo). Le sintassi possibili sono:

```
\path ... node[opzioni] (nome) {testo}  
node[opzioni] (nome) at (coordinate) {testo}
```

La seconda opzione è possibile solo al di fuori dell'ambiente `path` e permette di inserire il nodo alle coordinate specificate; in un ambiente `path` il nodo sarà inserito al punto corrente.

Nodi - esempio

```
\begin{tikzpicture}
\draw
(0,0)node[rectangle,draw]{A}(1,1)node[circle,draw]{B}
(2,0)node[draw]{C}(1,-1)node{D};
\end{tikzpicture}
```

B

A

C

D

Nodi - 2

Se non vengono date specificazioni sulla forma il testo è inserito in un rettangolo e, se non viene data l'opzione `draw` il nodo contiene solo il testo.

È possibile dare opzioni anche per il testo ed è possibile dire dove deve essere posto il nodo rispetto alle coordinate del punto:

- ▶ di default il nodo ha il suo centro nel punto in cui è posto
- ▶ `above`, `below`, `left` e `right` per spostarlo sopra, sotto, a sinistra o a destra
- ▶ le opzioni si possono combinare a due a due (`above left`)
- ▶ a ogni specificazione può essere aggiunta una misura come `above=3mm`

Nodi - esempio 2

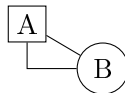
```
\begin{tikzpicture}
\fill (0,1) circle (1mm);
\draw (0,1)node[rectangle,draw,above]{A};
\fill (1,1) circle (1mm);
\draw (1,1)node[circle,draw,below
right=1mm]{B};
\fill (3,1) circle (1mm);
\draw (3,1)node[draw,left]{C};
\end{tikzpicture}
```



Nodi - esempio 3

Se a un nodo viene dato un nome è possibile successivamente usare quel nome per indicare il punto.

```
\begin{tikzpicture}
\draw (0,1)node(A)[draw,above]{A}(1,1)
node(B)[circle,draw,below]{B};
\draw (A)--(B);
\draw (B)-| (A);
\end{tikzpicture}
```

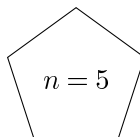
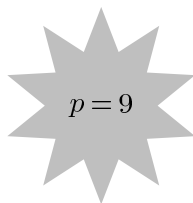


Nodi - 3

Caricando la libreria `shapes` con il comando `\usetikzlibrary{shapes}` è possibile usare altre forme per i nodi come stelle o poligoni regolari o ellissi; è inoltre possibile suddividere la regione interna di un nodo in parti in modo da avere del testo in ognuna di esse.

Nodi - esempio 4

```
\begin{tikzpicture}
\draw (0,0) node [star,star points=10,fill=lightgray] {$p=9$}
(3,0) node [regular polygon,regular polygon sides=5,draw]
{$n=5$}
(0,-2) node [ellipse,draw] {ellisse}
(3,-2) node [circle
split,draw,double]{sopra\nodepart{lower}{sotto}};
\end{tikzpicture}
```



II GUIT

Il primo riferimento in Italia per la documentazione su $\text{\LaTeX}$ è il sito del GUIT (Gruppo Italiano Utilizzatori $\text{\LaTeX}$)

-  **L'Arte di scrivere con $\text{\LaTeX}$** di Lorenzo Pantieri e Tommaso Gordini, un'introduzione a $\text{\LaTeX}$ rivolta sia ai principianti sia a utenti più avanzati
-  **$\text{\LaTeX}$ per l'impaziente** di Lorenzo Pantieri, è una versione ridotta dell'Arte
-  **Introduzione all'arte della composizione tipografica con $\text{\LaTeX}$** , di Claudio Beccari, è una guida più avanzata a $\text{\LaTeX}$ e alla tipografia digitale.
-  **$\text{\LaTeX}$ pedia**, di Lorenzo Pantieri, presenta $\text{\LaTeX}$ in modo più approfondito

Sempre sullo stesso sito sono a disposizione guide tematiche e molti numeri della rivista ArsTeXnica con altre guide ed esempi.

Guide per i pacchetti

Tutti i pacchetti $\text{\LaTeX}$ hanno la documentazione e cercando in rete si trovano esempi e suggerimenti.

Su *Overleaf* sono presenti guide ed esempi per i principali pacchetti

Problemi?

Per risolvere problemi, capire come si può fare qualcosa e tanto altro ci sono due forum a disposizione: il forum del GUIT

<https://www.guitex.org/home/it/forum/index> e

<https://tex.stackexchange.com/> (in inglese).