

# Prova di comandi Latex

Andrea Comar

October 2024

## Contents

<b>I</b>	<b>Titolo parte</b>	<b>2</b>
<b>1</b>	<b>Titolo sezione</b>	<b>2</b>
1.1	subsection . . . . .	2
<b>2</b>	<b>Titolo sezione</b>	<b>2</b>
2.1	titolo sottosezione . . . . .	2
<b>II</b>	<b>Cose utili</b>	<b>3</b>
<b>3</b>	<b>Nuovo documento</b>	<b>3</b>
<b>4</b>	<b>pacchetti utili</b>	<b>3</b>
<b>5</b>	<b>Comandi scrittura</b>	<b>4</b>
<b>6</b>	<b>Utilizzo delle tabelle</b>	<b>4</b>
6.1	ambiente tabular . . . . .	5
6.2	Esempio tabella . . . . .	6
<b>7</b>	<b>Simboli utili</b>	<b>7</b>
7.1	Simboli matematici . . . . .	7
<b>8</b>	<b>Utilizzo di algorithm2e</b>	<b>7</b>

## Part I

# Titolo parte

Questo è del testo a caso per far capire cosa succede se scrivi direttamente sotto una parte. La parte, con comando `\part` è la sezione più grossa in Latex. In automatico scrive parte I sopra.

## 1 Titolo sezione

La sezione è la seconda parte e puoi vedere il font automatico sopra queste righe di testo. comando `\section`

### 1.1 subsection

La sottosezione è questa parte, comando `\subsection`

## 2 Titolo sezione

Riutilizzando `section` conta in automatico quante sono le sezioni nel documento. Vale per tutti i comandi.

### 2.1 titolo sottosezione

**titolo paragrafo** Questo è un esempio di testo all'interno del paragrafo. Gli spazi sono automatici, non c'è bisogno di premere invio per andare a capo.

**titolo sottoparagrafo** In questo caso invece sto scrivendo un sottoparagrafo del precedente. Nel caso successivo provo a scrivere direttamente.

**secondo sottoparagrafo** questo è un ulteriore sottoparagrafo.  
Se vuoi cambiare pagina, devi usare `\newpage`

## Part II

# Cose utili

### 3 Nuovo documento

Per creare un nuovo documento, basta scrivere:

```
\documentclass{article} % tipo di documento
\usepackage{nome_pacchetto} % per i pacchetti

\title{titolo}
\author{autore}
\date{data}

\begin{document} % inizio del documento

\maketitle % per mostrare titolo, autore, data
\tableofcontents % per mostrare la tabella dei contenuti
...
\newpage % per cambiare pagina
...
\end{document}
```

La sintassi minima è la seguente

```
\documentclass{article}

\begin{document}
...
\end{document}
```

### 4 pacchetti utili

**graphicx** per inserire immagini

**amssymb** per i simboli matematici

**mathtools** per i simboli matematici

**algorithm2e** per scrivere algoritmi

**listings** per scrivere codice. si può scegliere un linguaggio oppure usarlo in modo simile al comando `\verbatim`. Consiglio la seguente impostazione:

```
\lstset{
  basicstyle=\ttfamily,
  mathescape
}
```

permette di scrivere i simboli matematici all'interno del codice, racchiundendoli con `$ ... $`

`color` per colorare il codice

## 5 Comandi scrittura

`\verbatim` Permette a Latex di non compilare il testo all'interno delle parentesi graffe.

```
if (a > b) {
  return a;
} else {
  return b;
}
```

`\begin{tabbing}` Permette di scrivere all'interno di una tabulazione. uso dello `\` per differenziare colonne.

```
\begin{tabbing}
  ciao\ ciao \ ciao
\end{tabbing}
```

ciao ciao ciao

`\textcolor{colore}{testo}` `testo rosso` ~ necessita del rispettivo pacchetto.

`\textit{testo}` *testo in corsivo*

`\textbf{testo}` **testo in grassetto**

`\underline{testo}` testo sottolineato

## 6 Utilizzo delle tabelle

In questa sezione capiremo come creare tabelle su latex. proviamo!

## 6.1 ambiente tabular

L'ambiente tabular si chiama con

```
\begin{tabular}{condizioni}  
...  
\end{tabular}
```

Parametri:

- parametri `l,c,r` per allineamento orizzontale
- parametro `p{larghezza}` per la larghezza. esclude parametri allineamento.
- parametro `|` per linee verticali
- parametro `@{...}` per spaziare le colonne

Costruzione tabella:

- `&` per separare le colonne
- `\\` per andare a capo
- `\hline` inserisce linea orizzontale
- `\cline{i-j}` inserisce linea orizzontale tra le colonne i e j
- `\multicolumn{ncol}{allineamentooriz}{...}` per unire n colonne

## 6.2 Esempio tabella

```
\begin{tabular}{|l|r|c|}
\hline
\multicolumn{3}{|c|}{all}\\
\hline
l & r & c \\
\hline
\end{tabular}
```

all		
l	r	c

```
\begin{tabular}{|l|r|c|}
\hline
\multicolumn{3}{|c|}{allineamento}\\
\hline
l & r & c \\
\hline
\end{tabular}
```

allineamento		
l	r	c

```
\begin{tabular}{|c|l|r|}
\hline
\multicolumn{3}{|c|}{allineamento}\\
\hline
c & l & r \\
\hline
\end{tabular}
```

allineamento		
c	l	r

```
\begin{tabular}{|r|r|r|}
\hline
\multicolumn{3}{|c|}{allineamento}\\
\hline
\multicolumn{3}{|c|}{testo di prova & prova & interpretazione}\\
\hline
\end{tabular}
```

allineamento		
testo di prova	prova	interpretazione

allineamento		
l	r	c
allineamento		
c	l	r

allineamento		
testo di prova	prova	interpretazione

Tabella di prova			
Nome	Cognome	Età	Sesso
Andrea	Comar	23	M
Stefano	Giorda	23	F

## 7 Simboli utili

### 7.1 Simboli matematici

Simboli matematici			
Simbolo	Comando	Simbolo	Comando
$\cup$	<code>\cup</code>	$\cap$	<code>\cap</code>
$\subset$	<code>\subset</code>	$\supseteq$	<code>\supseteq</code>
$\leftarrow$	<code>\leftarrow</code>	$\rightarrow$	<code>\rightarrow</code>
$\vdash$	<code>\vdash</code>	$\vDash$	<code>\vDash</code>
$\wedge$	<code>\wedge</code>	$\vee$	<code>\vee</code>

## 8 Utilizzo di algorithm2e

per iniziare a scrivere un algoritmo devo utilizzare il comando

`\begin{algorithm}[H]` e `\end{algorithm}` alla fine del codice.

con il comando `\SetAlgoLined` posso decidere se mettere o no le linee per separare le righe dell'algoritmo.

con il comando `\KwData` posso scrivere i dati in ingresso dell'algoritmo.

**Data:** this text

con il comando `\KwResult` posso scrivere i dati in uscita dell'algoritmo.

**Result:** how to write algorithm with L<sup>A</sup>T<sub>E</sub>X2e

**For** con il comando \For

```
for content... do  
| for-block  
end
```

**ForEach** con il comando \foreach

```
foreach condition do content...  
| foreach-block  
end
```

**while** con il comando \While

```
while content... do  
| while-block  
end
```



**If** con il comando `\If`

```
if content... then  
  | then-block  
end
```

con il comando `\elf` posso scrivere un if-else.

```
if condition then  
  | then-block  
else  
  | else-block  
end
```

**Else** con il comando `\Else` posso scrivere un else.

```
if Chiara è Piena then  
  | si slaccia la cintura;  
  | e si rompe il cazzo;  
else  
  | Si lamenta che ha fame  
end
```