

L^AT_EX for dummies

Guida di sopravvivenza per fisici

Lezione 1 - Le basi

Leonardo Pacciani-Mori

`leonardo.pacciani@phd.unipd.it`

AISF - Comitato Locale di Padova

6 dicembre 2018



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Quattro lezioni da due ore ciascuna. Argomenti:

- 1 Le basi
- 2 Matematica
- 3 Le classi report, article e book
- 4 La classe beamer

Quattro lezioni da due ore ciascuna. Argomenti:

- 1 Le basi
- 2 Matematica
- 3 Le classi report, article e book
- 4 La classe beamer

Scopo

Porre le basi per un uso e una comprensione di base di \LaTeX .

Competenze acquisite

Essere in grado di scrivere documenti con \LaTeX , quali relazioni di laboratorio, tesi e presentazioni.

Quattro lezioni da due ore ciascuna. Argomenti:

- 1 Le basi
- 2 Matematica
- 3 Le classi report, article e book
- 4 La classe beamer

Scopo

Porre le basi per un uso e una comprensione di base di \LaTeX .

Competenze acquisite

Essere in grado di scrivere documenti con \LaTeX , quali relazioni di laboratorio, tesi e presentazioni.

Referenza su cui si basa il corso: [L'arte di scrivere con \$\text{\LaTeX}\$](#) , di Lorenzo Pantieri.

Quattro lezioni da due ore ciascuna. Argomenti:

- 1 Le basi
- 2 Matematica
- 3 Le classi report, article e book
- 4 La classe beamer

Scopo

Porre le basi per un uso e una comprensione di base di \LaTeX .

Competenze acquisite

Essere in grado di scrivere documenti con \LaTeX , quali relazioni di laboratorio, tesi e presentazioni.

Referenza su cui si basa il corso: [L'arte di scrivere con \$\text{\LaTeX}\$](#) , di Lorenzo Pantieri.
Tutte le informazioni sul corso (comprese le slides) si trovano [qui](#).

In questa lezione



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- 1 Informazioni base su \LaTeX (senza excursus storico)
- 2 Esempio di documento “Hello world”, con commenti riga per riga
- 3 Alcuni comandi utili
- 4 Struttura generale di un documento
- 5 Tabelle e figure

Le basi

Che cos'è \LaTeX , e come funziona?



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Che cos'è \LaTeX , e come funziona?

- È un **linguaggio di markup**, che serve per la composizione tipografica di documenti di qualunque tipo

Che cos'è \LaTeX , e come funziona?

- È un **linguaggio di markup**, che serve per la composizione tipografica di documenti di qualunque tipo
- Il “ \TeX ” di \LaTeX proviene dalla radice greca “ $\tau\epsilon\chi$ ” (come in $\tau\epsilon\chi\nu\eta$), e quindi si pronuncia “tèch” (ma “tèk” è anche accettato)

Che cos'è \LaTeX , e come funziona?

- È un **linguaggio di markup**, che serve per la composizione tipografica di documenti di qualunque tipo
- Il “ \TeX ” di \LaTeX proviene dalla radice greca “ $\tau\epsilon\chi$ ” (come in $\tau\epsilon\chi\nu\eta$), e quindi si pronuncia “tèch” (ma “tèk” è anche accettato)
- È un linguaggio caratterizzato da **composizione asincrona**

Che cos'è \LaTeX , e come funziona?

- È un **linguaggio di markup**, che serve per la composizione tipografica di documenti di qualunque tipo
- Il “ \TeX ” di \LaTeX proviene dalla radice greca “ $\tau\epsilon\chi$ ” (come in $\tau\epsilon\chi\nu\eta$), e quindi si pronuncia “tèch” (ma “tèk” è anche accettato)
- È un linguaggio caratterizzato da **composizione asincrona**
Composizione sincrona: l'utente agisce direttamente su un testo già composto, che appare come verrà stampato. Ogni modifica si traduce in una variazione immediata di quel testo (es: word processors)

Che cos'è \LaTeX , e come funziona?

- È un **linguaggio di markup**, che serve per la composizione tipografica di documenti di qualunque tipo
- Il “ \TeX ” di \LaTeX proviene dalla radice greca “ $\tau\epsilon\chi$ ” (come in $\tau\epsilon\chi\nu\eta$), e quindi si pronuncia “tèch” (ma “tèk” è anche accettato)
- È un linguaggio caratterizzato da **composizione asincrona**
 - Composizione sincrona:** l'utente agisce direttamente su un testo già composto, che appare come verrà stampato. Ogni modifica si traduce in una variazione immediata di quel testo (es: word processors)
 - Composizione asinctrona:** l'utente inserisce il testo in un editor, concentrandosi su contenuto e struttura per poi darlo in pasto a un compositore, che lo impagina in modo ottimale

Le basi

Hello world



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Il codice sorgente di un documento scritto in \LaTeX è simile al sorgente di un programma: contiene le istruzioni che il compositore deve eseguire per dare l'output desiderato.

Il codice sorgente di un documento scritto in \LaTeX è simile al sorgente di un programma: contiene le istruzioni che il compositore deve eseguire per dare l'output desiderato.

“Hello world” di un documento in \LaTeX , confrontato con uno in C++:

<code>\documentclass[a4paper]{article}</code>	1	<code>//inizio preambolo</code>
<code>%inizio preambolo</code>	2	<code>#include<iostream></code>
<code>\usepackage[T1]{fontenc}</code>	3	<code>using namespace std;</code>
<code>\usepackage[utf8]{inputenc}</code>	4	
<code>\usepackage[italian]{babel}</code>	5	<code>//fine preambolo</code>
<code>%fine preambolo</code>	6	<code>int main(){</code>
<code>\begin{document}</code>	7	<code>//inizio corpo</code>
<code>%inizio corpo</code>	8	<code>cout << "Hello_\uworld!";</code>
<code>Hello world!</code>	9	<code>return 0;</code>
<code>%fine corpo</code>	10	<code>//fine corpo</code>
<code>\end{document}</code>	11	<code>}</code>

Hello world

Comandi e ambienti



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

I comandi \LaTeX iniziano **sempre** con un backslash \backslash , e hanno la seguente struttura:

$$\backslash \text{nomecomando} \underbrace{[\dots] \dots [\dots]}_{\text{argomenti facoltativi}} \underbrace{\{\dots\} \dots \{\dots\}}_{\text{argomenti obbligatori}}$$

I comandi \LaTeX iniziano **sempre** con un backslash `\`, e hanno la seguente struttura:

$$\backslash \text{nomecomando} \underbrace{[\dots] \dots [\dots]}_{\text{argomenti facoltativi}} \underbrace{\{\dots\} \dots \{\dots\}}_{\text{argomenti obbligatori}}$$

All'interno delle parentesi gli argomenti sono separati da virgole.

Se un comando non ha argomenti è detto **dichiarazione**, ed ha effetto sul testo dal punto in cui viene inserito fino alla fine del blocco (es.: `\itshape`).

I comandi \LaTeX iniziano **sempre** con un backslash `\`, e hanno la seguente struttura:

$$\backslash \text{nomecomando} \underbrace{[\dots] \dots [\dots]}_{\text{argomenti facoltativi}} \underbrace{\{\dots\} \dots \{\dots\}}_{\text{argomenti obbligatori}}$$

All'interno delle parentesi gli argomenti sono separati da virgole.

Se un comando non ha argomenti è detto **dichiarazione**, ed ha effetto sul testo dal punto in cui viene inserito fino alla fine del blocco (es.: `\itshape`).

Gli **ambienti** sono porzioni di codice delimitate da comandi di apertura e chiusura. Struttura generale:

```
\begin{ambiente} [\dots] \{\dots\}
  %corpo
\end{ambiente}
```

Esempi: `document`, `itemize`, `enumerate`. Gli ambienti possono essere annidati.

Hello world

Preambolo



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Il **preambolo** di un documento contiene:

- dichiarazioni per caricare pacchetti che estendono le capacità di \LaTeX (es.: inserire figure, fare disegni, simboli matematici ecc.), attraverso il comando `\usepackage[...]{nomepacchetto}`
- definizioni di comandi e aggrassettoambienti personalizzati
- opzioni generali del documento (font, margini ecc.)

Il **preambolo** di un documento contiene:

- dichiarazioni per caricare pacchetti che estendono le capacità di \LaTeX (es.: inserire figure, fare disegni, simboli matematici ecc.), attraverso il comando `\usepackage[...]{nomepacchetto}`
- definizioni di comandi e aggrassettoambienti personalizzati
- opzioni generali del documento (font, margini ecc.)

```
\documentclass[a4paper]{article} 1 //inizio preambolo
%inizio preambolo                2 #include<iostream>
\usepackage[T1]{fontenc}         3 using namespace std;
\usepackage[utf8]{inputenc}      4
\usepackage[italian]{babel}     5 //fine preambolo
%fine preambolo                  6 int main(){
\begin{document}                 7 //inizio corpo
%inizio corpo                    8 cout << "Hello_\uworld!";
Hello world!                     9 return 0;
%fine corpo                      10 //fine corpo
\end{document}                   11 }
```


Hello world

```
\documentclass[a4paper]{article}
```



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

```
\documentclass[a4paper]{article}
```

Il comando `documentclass` serve per dichiarare la **classe** del documento (articolo, libro, presentazione) e specificarne le dimensioni (foglio A4 ecc.).

```
\documentclass[a4paper]{article}
```

Il comando `documentclass` serve per dichiarare la **classe** del documento (articolo, libro, presentazione) e specificarne le dimensioni (foglio A4 ecc.).

Le principali classi predefinite sono:

- `article`, per articoli
- `report`, per relazioni
- `book`, per libri e tesi
- `letter`, per lettere intestate
- `beamer`, per presentazioni

```
\documentclass[a4paper]{article}
```

Il comando `documentclass` serve per dichiarare la **classe** del documento (articolo, libro, presentazione) e specificarne le dimensioni (foglio A4 ecc.).

Le principali classi predefinite sono:

- `article`, per articoli
- `report`, per relazioni
- `book`, per libri e tesi
- `letter`, per lettere intestate
- `beamer`, per presentazioni

Gli argomenti facoltativi sono globali per tutto il documento. Esempi:

- `10pt`, `11pt`, `12pt`, ...: dimensione principale del font
- `letterpaper`, `a4paper`, `a5paper`, ...: dimensione del foglio

Hello world

fontenc, inputenc e babel



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

```
\documentclass[a4paper]{article} 1 //inizio preambolo
%inizio preambolo                2 #include<iostream>
\usepackage[T1]{fontenc}         3 using namespace std;
\usepackage[utf8]{inputenc}      4
\usepackage[italian]{babel}     5
%fine preambolo                  6 //fine preambolo
\begin{document}                7 int main(){
%inizio corpo                    8 //inizio corpo
Hello world!                     9 cout << "Hello_\uworld!";
%fine corpo                     10 return 0; //fine corpo
\end{document}                  11 }
```

Sono direttive necessarie per poter scrivere in italiano (o qualsiasi altra lingua). Rimuovendole è comunque possibile scrivere in inglese.

fontenc: *font encoding*, serve per gestire il font del documento; T1 è la codifica per le lingue occidentali

inputenc: *input encoding*, serve per interpretare correttamente i caratteri speciali (lettere accentate ecc.); utf8 è la codifica generalmente usata per interpretare gli input da tastiera

babel: gestisce parole fisse (generate automaticamente da alcuni comandi, come “capitolo”, “appendice”, “indice” e il formato della data)

Hello world



fontenc, inputenc e babel

```
\documentclass[a4paper]{article} 1 //inizio preambolo
%inizio preambolo                2 #include<iostream>
\usepackage[T1]{fontenc}         3 using namespace std;
\usepackage[utf8]{inputenc}      4
\usepackage[italian]{babel}     5
%fine preambolo                  6 //fine preambolo
\begin{document}                 7 int main(){
%inizio corpo                     8 //inizio corpo
Hello world!                      9 cout << "Hello_\uworld!";
%fine corpo                      10 return 0; //fine corpo
\end{document}                   11 }
```

`\begin{document}` e `\end{document}` delimitano, rispettivamente, l'inizio e la fine del contenuto del documento. Tutto ciò che nel sorgente viene scritto dopo `\end{document}` viene ignorato dal compilatore.

A seconda della sua classe, un documento può essere diviso in più parti:

- `part`
- `chapter`
- `section`
- `subsection`
- `subsubsection`
- `paragraph`
- `subparagraph`

Ognuno di questi è un comando che prende come argomento obbligatorio il nome della parte/capitolo/sezione ecc.

A seconda della sua classe, un documento può essere diviso in più parti:

- `part` Ad esempio, `\chapter{La relatività ristretta}` in un documento `book` restituisce:
Capitolo 1
- `chapter` **La relatività ristretta**
- `section` **La relatività ristretta**
- `subsection` Aggiungendo `\section{L'esperimento di Michelson e Morley}`:
Capitolo 1
- `subsubsection` **La relatività ristretta**
- `paragraph` **La relatività ristretta**
- `subparagraph` 1.1 L'esperimento di Michelson e Morley

Ognuno di questi è un comando che prende come argomento obbligatorio il nome della parte/capitolo/sezione ecc.

Il compilatore \LaTeX gestisce **automaticamente** la numerazione di tutti gli elementi di un documento: pagine, capitoli/sezioni, equazioni, figure, tabelle, referenze bibliografiche ecc.

Il compilatore \LaTeX gestisce **automaticamente** la numerazione di tutti gli elementi di un documento: pagine, capitoli/sezioni, equazioni, figure, tabelle, referenze bibliografiche ecc.

L'indice di un documento viene creato automaticamente col comando `\tableofcontents`. Analogamente, `\listoffigures` e `\listoftables` generano gli indici di figure e tabelle (mostrando le didascalie).

Il compilatore \LaTeX gestisce **automaticamente** la numerazione di tutti gli elementi di un documento: pagine, capitoli/sezioni, equazioni, figure, tabelle, referenze bibliografiche ecc.

L'indice di un documento viene creato automaticamente col comando `\tableofcontents`. Analogamente, `\listoffigures` e `\listoftables` generano gli indici di figure e tabelle (mostrando le didascalie).

Come vedremo più avanti, poi, ci sono più modi di creare la bibliografia di un documento.

Alcuni comandi utili

Modificare lo stile e le dimensioni del testo



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Alcuni comandi utili



Modificare lo stile e le dimensioni del testo

Lo stile del testo può essere modificato coi seguenti comandi:

Testo *corsivo*: `\textit{...}` o `\itshape`

Testo "*enfaticizzato*": `\emph{...}` o `\em`

Testo **grassetto**: `\textbf{...}` o `\bfseries`

Testo MAIUSCOLETTA: `\textsc{...}` o `\scshape`

Testo "macchina da scrivere": `\texttt{...}` o `\ttfamily`

Alcuni comandi utili



Modificare lo stile e le dimensioni del testo

Lo stile del testo può essere modificato coi seguenti comandi:

Testo *corsivo*: `\textit{...}` o `\itshape`

Testo ***enfaticizzato***: `\emph{...}` o `\em`

Testo **grassetto**: `\textbf{...}` o `\bfseries`

Testo MAIUSCOLETTA: `\textsc{...}` o `\scshape`

Testo "macchina da scrivere": `\texttt{...}` o `\ttfamily`

Le dimensioni, invece, coi seguenti comandi:

`\tiny` `\scriptsize` `\footnotesize` `\small` `\normalsize` `\large`

`\Large` `\LARGE` `\huge` `\Huge`

Alcuni comandi utili

Colori, note a margine e a piè pagina



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Alcuni comandi utili



Colori, note a margine e a piè pagina

Il colore di una parte di testo può essere cambiato usando il pacchetto `color`, e la sintassi del comando per cambiare colore è:

`\textcolor{colore}{testo}` oppure `{\color{colore}testo}`

I colori predefiniti nel pacchetto sono:

`black`, `blue`, `brown`, `cyan`, `darkgray`, `gray`, `green`, `lightgray`, `lime`,
`magenta`, `olive`, `orange`, `pink`, `purple`, `red`, `teal`, `violet`, `white`, `yellow`

Alcuni comandi utili



Colori, note a margine e a piè pagina

Il colore di una parte di testo può essere cambiato usando il pacchetto `color`, e la sintassi del comando per cambiare colore è:

`\textcolor{colore}{testo}` oppure `{\color{colore}testo}`

I colori predefiniti nel pacchetto sono:

`black`, `blue`, `brown`, `cyan`, `darkgray`, `gray`, `green`, `lightgray`, `lime`,
`magenta`, `olive`, `orange`, `pink`, `purple`, `red`, `teal`, `violet`, `white`, `yellow`

Le note a margine di un testo si inseriscono caricando il pacchetto `marginnote` e usando il comando `\marginpar`. Le note a piè pagina, invece, si inseriscono col comando `\footnote` (senza bisogno di caricare pacchetti particolari).

Alcuni comandi utili

Elenchi puntati e numerati



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Alcuni comandi utili



Elenchi puntati e numerati

Gli elenchi puntati si fanno con l'ambiente `itemize`, e ogni punto è contrassegnato dal comando `\item`. Ad esempio:

Codice:

```
\begin{itemize}  
  \item primo punto  
  \item secondo punto  
\end{itemize}
```

Risultato:

- primo punto
- secondo punto

Alcuni comandi utili



Elenchi puntati e numerati

Gli elenchi puntati si fanno con l'ambiente `itemize`, e ogni punto è contrassegnato dal comando `\item`. Ad esempio:

Codice:

```
\begin{itemize}  
  \item primo punto  
  \item secondo punto  
\end{itemize}
```

Risultato:

- primo punto
- secondo punto

Analogamente, gli elenchi numerati si creano con l'ambiente `enumerate`:

Codice:

```
\begin{enumerate}  
  \item primo punto  
  \item secondo punto  
\end{enumerate}
```

Risultato:

- 1 primo punto
- 2 secondo punto

Elenchi puntati e numerati

Gli elenchi puntati si fanno con l'ambiente `itemize`, e ogni punto è contrassegnato dal comando `\item`. Ad esempio:

Codice:

```
\begin{itemize}
  \item primo punto
  \item secondo punto
\end{itemize}
```

Risultato:

- primo punto
- secondo punto

Analogamente, gli elenchi numerati si creano con l'ambiente `enumerate`:

Codice:

```
\begin{enumerate}
  \item primo punto
  \item secondo punto
\end{enumerate}
```

Risultato:

- 1 primo punto
- 2 secondo punto

L'ambiente `description` permette di personalizzare i vari punti con una “descrizione”:

Codice:

```
\begin{description}
  \item[descrizione 1] primo punto
  \item[descrizione 2] secondo punto
\end{description}
```

Risultato:

- descrizione 1 primo punto
- descrizione 2 secondo punto

Sono oggetti **floating**, ossia \LaTeX si occupa di posizionarli al meglio all'interno del documento. La struttura del codice per le tabelle è:

```
\begin{table}[preferenze di posizionamento]  
...  
\end{table}
```

mentre per le figure è:

```
\begin{figure}[preferenze di posizionamento]  
...  
\end{figure}
```

Ognuno di questi oggetti riceve automaticamente un numero progressivo di identificazione.

Sono oggetti **floating**, ossia \LaTeX si occupa di posizionarli al meglio all'interno del documento. La struttura del codice per le tabelle è:

```
\begin{table}[preferenze di posizionamento]  
...  
\end{table}
```

mentre per le figure è:

```
\begin{figure}[preferenze di posizionamento]  
...  
\end{figure}
```

Ognuno di questi oggetti riceve automaticamente un numero progressivo di identificazione.

La didascalia si aggiunge usando il pacchetto `caption` e il comando `\caption{...}` all'interno dell'ambiente `table` o `figure`; usando `\caption*{...}` la figura non riceve un numero di identificazione. Argomento facoltativo: versione "ridotta" della didascalia (per `listoffigures` e `listoftables`).

Preferenze di posizionamento:

- h (*here*, cioè dove si trova nel sorgente)
- t (*top*, all'inizio della pagina)
- b (*bottom*, al fondo della pagina)
- p (*page of floats*, ossia in una pagina dedicata)

Preferenze di posizionamento:

- `h` (*here*, cioè dove si trova nel sorgente)
- `t` (*top*, all'inizio della pagina)
- `b` (*bottom*, al fondo della pagina)
- `p` (*page of floats*, ossia in una pagina dedicata)

Le preferenze si possono giustapporre in ordine di priorità (ad es.: `htb`). Se \LaTeX non riesce a soddisfare le nostre richieste, prova a posizionare la figura nella pagina successiva e così via.

Preferenze di posizionamento:

- `h` (*here*, cioè dove si trova nel sorgente)
- `t` (*top*, all'inizio della pagina)
- `b` (*bottom*, al fondo della pagina)
- `p` (*page of floats*, ossia in una pagina dedicata)

Le preferenze si possono giustapporre in ordine di priorità (ad es.: `htb`). Se \LaTeX non riesce a soddisfare le nostre richieste, prova a posizionare la figura nella pagina successiva e così via.

Attenzione

È **sempre** preferibile lasciare che \LaTeX posizioni le figure e le tabelle dove ritiene più opportuno, seguendo le nostre preferenze di posizionamento.

Preferenze di posizionamento:

- `h` (*here*, cioè dove si trova nel sorgente)
- `t` (*top*, all'inizio della pagina)
- `b` (*bottom*, al fondo della pagina)
- `p` (*page of floats*, ossia in una pagina dedicata)

Le preferenze si possono giustapporre in ordine di priorità (ad es.: `htb`). Se \LaTeX non riesce a soddisfare le nostre richieste, prova a posizionare la figura nella pagina successiva e così via.

Attenzione

È **sempre** preferibile lasciare che \LaTeX posizioni le figure e le tabelle dove ritiene più opportuno, seguendo le nostre preferenze di posizionamento.

Nel caso in cui si voglia forzare \LaTeX a posizionare una tabella o figura dove l'abbiamo messa all'interno del codice sorgente, si può usare il pacchetto `float` e la preferenza di posizionamento `H`.

La struttura generale di una tabella è:

```
\begin{table}  
  \centering  
  \begin{tabular}{allineamento}  
    %corpo  
  \end{tabular}  
  \caption{...}  
\end{table}
```

La struttura generale di una tabella è:

```
\begin{table}  
  \centering  
  \begin{tabular}{allineamento}  
    %corpo  
  \end{tabular}  
  \caption{...}  
\end{table}
```

Il corpo di una tabella si scrive *riga per riga*, separando le colonne con & e le nuove righe con `\\`, le linee verticali con `|` e quelle orizzontali con `\hline`. Usando il pacchetto `booktabs` si possono fare linee di divisione “professionali”: `\toprule`, `\midrule` e `\bottomrule`.

La struttura generale di una tabella è:

```
\begin{table}  
  \centering  
  \begin{tabular}{allineamento}  
    %corpo  
  \end{tabular}  
  \caption{...}  
\end{table}
```

Il corpo di una tabella si scrive *riga per riga*, separando le colonne con & e le nuove righe con `\\`, le linee verticali con `|` e quelle orizzontali con `\hline`. Usando il pacchetto `booktabs` si possono fare linee di divisione “professionali”: `\toprule`, `\midrule` e `\bottomrule`.

La voce `allineamento` indica come il testo deve essere allineato all'interno delle celle, e deve contenere per ogni colonna una delle seguenti lettere: `c` (center), `l` (left), `r` (right).

Primo esempio:

Codice:

```
\begin{table}  
\centering  
\begin{tabular}{cc}  
Prima colonna & seconda colonna \\  
\hline  
prima cella & seconda cella \\  
\hline  
terza cella & quarta cella  
\end{tabular}  
\caption{Tabella di esempio}  
\end{table}
```

Risultato:

Prima colonna	seconda colonna
prima cella	seconda cella
terza cella	quarta cella

Tabella: Tabella di esempio

Secondo esempio:

Codice:

```
\begin{table}  
\centering  
\begin{tabular}{|l|r|}  
Prima colonna & seconda colonna \\  
\hline  
prima cella & seconda cella \\  
\hline  
terza cella & quarta cella \\  
\hline  
\end{tabular}  
\caption{Altra tabella}  
\end{table}
```

Risultato:

Prima colonna	seconda colonna
prima cella	seconda cella
terza cella	quarta cella

Tabella: Altra tabella

Esempio finale (booktabs):

Codice:

```
\begin{table}  
\centering  
\begin{tabular}{cc}  
Prima colonna & seconda colonna \\  
\toprule  
prima cella & seconda cella \\  
\midrule  
terza cella & quarta cella \\  
\midrule  
quinta cella & sesta cella \\  
\bottomrule  
\end{tabular}  
\caption{Utilizzo di booktabs}  
\end{table}
```

Risultato:

Prima colonna	seconda colonna
prima cella	seconda cella
terza cella	quarta cella
quinta cella	sesta cella

Tabella: Utilizzo di booktabs

La struttura generale di una figura è:

```
\begin{figure}  
  \centering  
  \includegraphics[dimensioni]{file}  
  \caption{...}  
\end{figure}
```

Per poter utilizzare l'ambiente `figure` è necessario includere il pacchetto `graphicx`.

La struttura generale di una figura è:

```
\begin{figure}  
  \centering  
  \includegraphics[dimensioni]{file}  
  \caption{...}  
\end{figure}
```

Per poter utilizzare l'ambiente `figure` è necessario includere il pacchetto `graphicx`.

All'argomento `dimensioni` vanno sostituite le indicazioni sulle dimensioni della figura; si possono ad esempio indicare `width` e `height` (indicandone solo una \LaTeX mantiene automaticamente il loro rapporto costante), oppure `scale` per riscalare direttamente tutta l'immagine, e `angle` per ruotarla (in gradi, in senso antiorario).

La struttura generale di una figura è:

```
\begin{figure}  
  \centering  
  \includegraphics[dimensioni]{file}  
  \caption{...}  
\end{figure}
```

Per poter utilizzare l'ambiente `figure` è necessario includere il pacchetto `graphicx`.

All'argomento `dimensioni` vanno sostituite le indicazioni sulle dimensioni della figura; si possono ad esempio indicare `width` e `height` (indicandone solo una \LaTeX mantiene automaticamente il loro rapporto costante), oppure `scale` per riscalare direttamente tutta l'immagine, e `angle` per ruotarla (in gradi, in senso antiorario).

All'argomento `file` bisogna inserire la posizione del file immagine, rispetto alla cartella dove si trova il file `.tex` compilato.

Formati accettati: `pdf`, `jpg`, `bmp`, `png`, `eps`.

Primo esempio:

Risultato:

Codice:

```
\begin{figure}  
\centering  
\includegraphics[scale=0.5]{Einstein.jpg}  
\caption{Einstein nel 1947}  
\end{figure}
```

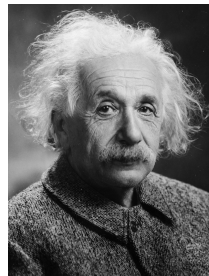


Figura: Einstein nel 1947

Secondo esempio:

Risultato:

Codice:

```
\begin{figure}  
  \centering  
  \includegraphics%  
    [scale=0.5,angle=45]{Einstein.jpg}  
  \caption{Einstein nel 1947, ruotato}  
\end{figure}
```

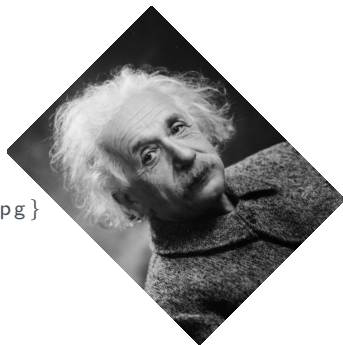


Figura: Einstein nel 1947, ruotato

Terzo esempio:

Risultato:

Codice:

```
\begin{figure}  
\centering  
\includegraphics%  
[width=50px,height=120px]{Einstein.jpg}  
\caption{Einstein nel 1947, stiracchiato}  
\end{figure}
```

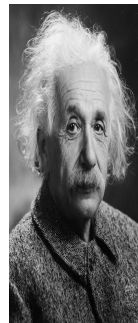


Figura: Einstein nel 1947, stiracchiato

Quarto esempio:

Codice:

```
\begin{figure}  
\centering  
\includegraphics%  
    [width=50px]{Einstein.jpg}  
\caption{Einstein nel 1947, normale}  
\end{figure}
```

Risultato:

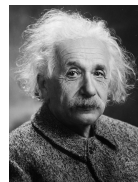


Figura: Einstein nel 1947, normale

Ultimo esempio:

Risultato:

Codice:

```
\begin{figure}  
\centering  
\includegraphics%  
    [height=120px]{Einstein.jpg}  
\caption{Einstein nel 1947, normale}  
\end{figure}
```

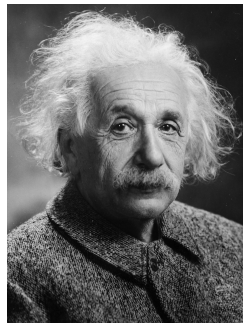


Figura: Einstein nel 1947, normale

Per oggi è tutto!

Prossimo appuntamento: 10 dicembre 2018, **aula LUF2**
ore 16:30

Prossima lezione: Matematica



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Cambio orario e aule

Rispetto a quanto annunciato ci sono stati dei cambi di aule e orario per le prossime lezioni, che saranno:

- il 10 dicembre in aula **LUF2** dalle 16:30 alle 18:30
- il 12 dicembre in aula LUF2 **dalle 14:30 alle 16:30**
- il 13 dicembre in aula **LUF2** dalle 16:30 alle 18:30

Un po' di (sano) spam



Dal 13 dicembre si apriranno le iscrizioni per la **V Conferenza Italiana degli Studenti di Fisica**, che si terrà **a Milano dal 6 al 10 marzo 2019**.

Dal 13 dicembre si apriranno le iscrizioni per la **V Conferenza Italiana degli Studenti di Fisica**, che si terrà a **Milano dal 6 al 10 marzo 2019**.



Per maggiori informazioni: ai-sf.it/cisf19