

Fondamenti del linguaggio \LaTeX per la scrittura scientifica

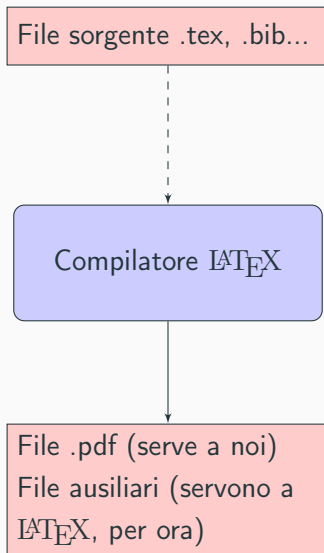
Lezione 1: Introduzione e basi del linguaggio

Andrea Di Primio (andrea.diprimio@polimi.it)

29 Novembre 2021

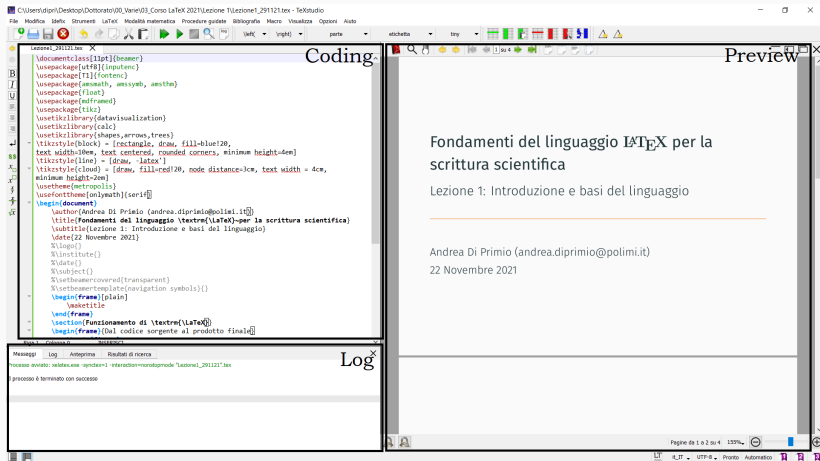
Funzionamento di L^AT_EX

Dal codice sorgente al prodotto finale








L'editor \TeX Studio: Struttura dell'interfaccia utente






Aprendo l'editor, ci troviamo davanti un'interfaccia tripartita:








Di seguito, i simboli che useremo più spesso (per il momento ignoreremo gli altri):

-  Crea una nuova istanza di scrittura codice.
-  Apre un file .tex (o altro formato L^AT_EX) esistente.
-  Salva il file corrente.
-  Compila il codice corrente e visualizza in Preview il file output .pdf (shortcut F1 o F5).
-  Ferma la compilazione (attivo solo durante la compilazione).






Di seguito, i simboli che useremo più spesso (per il momento ignoreremo gli altri):

-  Crea una nuova istanza di scrittura codice.
-  Apre un file .tex (o altro formato L^AT_EX) esistente.
-  Salva il file corrente.
-  Compila il codice corrente e visualizza in Preview il file output .pdf (shortcut F1 o F5).
-  Ferma la compilazione (attivo solo durante la compilazione).






Di seguito, i simboli che useremo più spesso (per il momento ignoreremo gli altri):

-  Crea una nuova istanza di scrittura codice.
-  Apre un file .tex (o altro formato L^AT_EX) esistente.
-  Salva il file corrente.
-  Compila il codice corrente e visualizza in Preview il file output .pdf (shortcut F1 o F5).
-  Ferma la compilazione (attivo solo durante la compilazione).

Di seguito, i simboli che useremo più spesso (per il momento ignoreremo gli altri):

-  Crea una nuova istanza di scrittura codice.
-  Apre un file .tex (o altro formato L^AT_EX) esistente.
-  Salva il file corrente.
-  Compila il codice corrente e visualizza in Preview il file output .pdf (shortcut F1 o F5).
-  Ferma la compilazione (attivo solo durante la compilazione).

Di seguito, i simboli che useremo più spesso (per il momento ignoreremo gli altri):

-  Crea una nuova istanza di scrittura codice.
-  Apre un file .tex (o altro formato L^AT_EX) esistente.
-  Salva il file corrente.
-  Compila il codice corrente e visualizza in Preview il file output .pdf (shortcut F1 o F5).
-  Ferma la compilazione (attivo solo durante la compilazione).

Programmazione in L^AT_EX

Indipendentemente dal tipo di progetto, un file .tex è, in generale, composto da due parti:

- il **preambolo**, contenente direttive generalmente relative al codice \LaTeX , e.g. inclusione di pacchetti di comandi, definizione di comandi personalizzati.
- il **corpo**, contenente i contenuti del documento, e.g. i capitoli di un libro, le appendici e la bibliografia.

Indipendentemente dal tipo di progetto, un file .tex è, in generale, composto da due parti:

- il **preambolo**, contenente direttive generalmente relative al codice \LaTeX , e.g. inclusione di pacchetti di comandi, definizione di comandi personalizzati.
- il **corpo**, contenente i contenuti del documento, e.g. i capitoli di un libro, le appendici e la bibliografia.

Tipologie di direttive L^AT_EX: comandi

Le direttive in codice L^AT_EX rientrano, grossomodo, in due classi. I **comandi** seguono il paradigma

```
\commandname [options]{arg1}{arg2}...{argn}
```

Ad esempio, il comando

```
\documentclass [10pt, a4paper]{article}
```

ha due argomenti opzionali (tra quadre) e uno obbligatorio (tra graffe). Alcuni comandi potrebbero avere zero argomenti (opzionali e/o obbligatori).

Tipologie di direttive \LaTeX : ambienti

Gli **ambienti**, invece, sono "container" di codice delimitati da un inizio e una fine:

```
\begin{envname}[options]{arg1}{arg2}...{argn}  
    contenuto...  
\end{envname}
```

Ad esempio, l'ambiente

```
\begin{itemize}  
    contenuto...  
\end{itemize}
```

struttura un elenco di oggetti.

Cominciamo a programmare!

Create un nuovo file su T_EXStudio e salvatelo in una cartella dedicata come `Lezione1.tex`.

Durante la lezione, lavorate **solo** su questo file. Alla fine, avrete generato un piccolo *cheatsheet*!

Codice del preambolo, parte 1: documentclass

```
\documentclass[options]{classname}
```

Le classi principali sono:

- `article` per articoli scientifici o brevi elaborati;
- `report` per lunghi elaborati (come una tesi);
- `book` per libri;
- `beamer` per presentazioni.

Le opzioni dipendono molto dalla classe scelta. Intanto ne citiamo due:

- `a4paper` per impostare il formato del foglio;
- `10pt`, `11pt` o `12pt` per impostare la dimensione del font.

Codice del preambolo, parte 1: documentclass

```
\documentclass[options]{classname}
```

Le classi principali sono:

- `article` per articoli scientifici o brevi elaborati;
- `report` per lunghi elaborati (come una tesi);
- `book` per libri;
- `beamer` per presentazioni.

Le opzioni dipendono molto dalla classe scelta. Intanto ne citiamo due:

- `a4paper` per impostare il formato del foglio;
- `10pt`, `11pt` o `12pt` per impostare la dimensione del font.

Codice del preambolo, parte 1: documentclass

Inserite questo comando nella prima riga del vostro nuovo file:

```
\documentclass[a4paper, 10pt]{article}
```

Codice del preambolo, parte 2: pacchetti

```
\usepackage[options]{packagename}
```

Some useful packages:

- babel with italian option provides the Italian spelling;
- inputenc with utf8 option for accents and special characters;
- amsmath, amssymb, amsthm for math-related things;
- geometry to set margins;
- graphicx to handle figures and graphs;

Google is your friend

Esistono migliaia di pacchetti. Tra questi, c'è anche quello che vi servirà in futuro. Cercatelo!

Codice del preambolo, parte 2: pacchetti

```
\usepackage[options]{packagename}
```

Some useful packages:

- babel with italian option provides the Italian spelling;
- inputenc with utf8 option for accents and special characters;
- amsmath, amssymb, amsthm for math-related things;
- geometry to set margins;
- graphicx to handle figures and graphs;

Google is your friend

Esistono migliaia di pacchetti. Tra questi, c'è anche quello che vi servirà in futuro. Cercatelo!

Aggiungiamo qualche pacchetto al nostro documento, direttamente sotto alla `\documentclass`.

```
\usepackage[italian]{babel}  
\usepackage{amsmath, amsymb, amsthm}  
\usepackage[top=2cm, bottom=2cm, left=2cm, right  
            =2cm]{geometry}
```

Codice del preambolo, parte 3: frontespizio

I seguenti comandi impostano i dati da mostrare sul frontespizio del lavoro:

- `\author{names}` imposta l'autore;
- `\date{\today}` imposta la data;
- `\title{nameTitle}` imposta il titolo.

Aggiungiamo al nostro documento i comandi:

```
\author{Il mio nome}  
\date{\today}  
\title{Lezione 1: Comandi base di \LaTeX}
```

Codice del preambolo, parte 3: frontespizio

I seguenti comandi impostano i dati da mostrare sul frontespizio del lavoro:

- `\author{names}` imposta l'autore;
- `\date{\today}` imposta la data;
- `\title{nameTitle}` imposta il titolo.

Aggiungiamo al nostro documento i comandi:

```
\author{Il mio nome}  
\date{\today}  
\title{Lezione 1: Comandi base di \LaTeX}
```

Codice del corpo: l'ambiente document

Il corpo del documento è, per definizione, tutto ciò che è inserito nell'ambiente document:

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

All'interno di questo ambiente potranno essere inseriti testo, comandi ed ambienti.

Warning!

Tutto ciò che segue `\end{document}` viene ignorato dal compilatore!

Codice del corpo: l'ambiente document

Il corpo del documento è, per definizione, tutto ciò che è inserito nell'ambiente document:

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

All'interno di questo ambiente potranno essere inseriti testo, comandi ed ambienti.

Warning!

Tutto ciò che segue `\end{document}` viene ignorato dal compilatore!

Esercizio 1: Hello, World!

Completiamo il nostro primo file e compiliamolo con F5.

```
\documentclass[a4paper, 10pt]{article}
\usepackage[italian]{babel}
\usepackage{amsmath, amsymb, amsthm}
\usepackage[top=2cm, bottom=2cm, left=2cm, right
    =2cm]{geometry}
\author{Il mio nome}
\date{\today}
\title{Lezione 1: Comandi base di \LaTeX}
\begin{document}
    \maketitle
    Hello, World!
\end{document}
```

Si usa suddividere un documento in parti più piccole.

<code>\section{sectionTitle}</code>	Crea una nuova sezione.
<code>\subsection{subsecTitle}</code>	Crea una nuova sottosezione.
<code>\chapter{chapterTitle}</code>	Crea un nuovo capitolo.

I capitoli sono riservati alle classi `report` e `book`.

Le sezioni vengono automaticamente numerate. Per creare una parte senza numerarla, aggiungere un asterisco al nome del comando, e.g. `\section*{sectionTitle}` o `\chapter*{sectionTitle}`.

Esercizio 2 (parte 1): Strutturare un documento.

```
% questo e' un commento!  
\begin{document}  
    \maketitle  
    \section{Esercizio 1: Hello, World!}  
    Hello, World!  
    \section{Esercizio 2: Strutturare un  
        documento}  
    % TO DO: creare una sottosezione  
        numerata.  
    % TO DO: creare una seconda sottosezione  
        non numerata.  
    \section{Esercizio 3: ?}  
    % ...  
\end{document}
```

Si può inserire testo **nel corpo** semplicemente scrivendolo, senza bisogno di comandi. Citiamo qualche comando per la formattazione.

`\textbf{textBold}` genera **textBold** (shortcut Ctrl + B)

`\textit{textIta}` genera *text/ta* (shortcut Ctrl + I)

`\texttt{textTt}` genera textTt

`\verb!verbatim!` genera verbatim, ma il contenuto viene ignorato dal compilatore (usato per scrivere codice L^AT_EX senza che il compilatore lo interpreti)

Si può inserire testo **nel corpo** semplicemente scrivendolo, senza bisogno di comandi. Citiamo qualche comando per la formattazione.

`\textbf{textBold}` genera **textBold** (shortcut Ctrl + B)

`\textit{textIta}` genera *text/ta* (shortcut Ctrl + I)

`\texttt{textTt}` genera `textTt`

`\verb!verbatim!` genera `verbatim`, ma il contenuto viene ignorato dal compilatore (usato per scrivere codice `LATEX` senza che il compilatore lo interpreti)

Si può inserire testo **nel corpo** semplicemente scrivendolo, senza bisogno di comandi. Citiamo qualche comando per la formattazione.

`\textbf{textBold}` genera **textBold** (shortcut Ctrl + B)

`\textit{textIta}` genera *textIta* (shortcut Ctrl + I)

`\texttt{textTt}` genera textTt

`\verb!verbatim!` genera verbatim, ma il contenuto viene ignorato dal compilatore (usato per scrivere codice L^AT_EX senza che il compilatore lo interpreti)

Si può inserire testo **nel corpo** semplicemente scrivendolo, senza bisogno di comandi. Citiamo qualche comando per la formattazione.

<code>\textbf{textBold}</code>	genera textBold (shortcut Ctrl + B)
<code>\textit{textIta}</code>	genera <i>textIta</i> (shortcut Ctrl + I)
<code>\texttt{textTt}</code>	genera textTt
<code>\verb!verbatim!</code>	genera verbatim, ma il contenuto viene ignorato dal compilatore (usato per scrivere codice L ^A T _E X senza che il compilatore lo interpreti)

Testo, parte 2: il comando `\\` e spacing del testo

Scrivendo testo, andare a capo nel codice **non** equivale ad iniziare una nuova riga.

Usare il comando `\\[distance]` per andare a capo lasciando *distance* di spazio bianco. Non specificare *distance* equivale ad andare a capo.

Si può specificare *distance* nei seguenti modi:

- espressa in un unità di misura (la scelta più comoda è cm o mm), ad esempio `\\[3mm]` o `\\[0.5cm]`;
- espressa come frazione di una lunghezza *macro*.

Testo, parte 2: il comando `\\` e spacing del testo

Scrivendo testo, andare a capo nel codice **non** equivale ad iniziare una nuova riga.

Usare il comando `\\[distance]` per andare a capo lasciando *distance* di spazio bianco. Non specificare *distance* equivale ad andare a capo.

Si può specificare *distance* nei seguenti modi:

- espressa in un unità di misura (la scelta più comoda è cm o mm), ad esempio `\\[3mm]` o `\\[0.5cm]`;
- espressa come frazione di una lunghezza *macro*.

Testo, parte 2: il comando `\\` e spacing del testo

Scrivendo testo, andare a capo nel codice **non** equivale ad iniziare una nuova riga.

Usare il comando `\\[distance]` per andare a capo lasciando *distance* di spazio bianco. Non specificare *distance* equivale ad andare a capo.

Si può specificare *distance* nei seguenti modi:

- espressa in un unità di misura (la scelta più comoda è cm o mm), ad esempio `\\[3mm]` o `\\[0.5cm]`;
- espressa come frazione di una lunghezza *macro*.

Digressione: lunghezze *macro*

Per aiutare nell'uniformare le spaziature, \LaTeX mette a disposizione dei comandi particolari (chiamate lunghezze *macro*), che rappresentano delle lunghezze "universali". Non richiedono alcun argomento: si chiamano semplicemente con `\macroname`.

Per le direttive che richiedono lunghezze, è buona norma cercare di riferirsi ad esse scrivendo `q\macroname`, dove `q` è un qualunque float (e.g. `0.5\macroname` indica metà di `\macroname`). *Spoiler: tuttavia, poter impostare lunghezze arbitrarie è uno strumento molto utile in alcune circostanze...*

Digressione: lunghezze *macro*

Per aiutare nell'uniformare le spaziature, \LaTeX mette a disposizione dei comandi particolari (chiamate lunghezze *macro*), che rappresentano delle lunghezze "universali". Non richiedono alcun argomento: si chiamano semplicemente con `\macroname`.

Per le direttive che richiedono lunghezze, è buona norma cercare di riferirsi ad esse scrivendo `q\macroname`, dove `q` è un qualunque float (e.g. `0.5\macroname` indica metà di `\macroname`). *Spoiler: tuttavia, poter impostare lunghezze arbitrarie è uno strumento molto utile in alcune circostanze...*

Testo, parte 2: il comando `\\` e spacing del testo

Una macro particolarmente utile per andare a capo è `\baselineskip`, che rappresenta l'altezza di una riga di testo (tenendo conto della grandezza del font).

Per andare a capo lasciando una riga bianca, possiamo scrivere `\\[\baselineskip]`.

Testo, parte 2: il comando `\\` e spacing del testo

Una macro particolarmente utile per andare a capo è `\baselineskip`, che rappresenta l'altezza di una riga di testo (tenendo conto della grandezza del font).

Per andare a capo lasciando una riga bianca, possiamo scrivere `\\[\baselineskip]`.

Esercizio 2 (parte 2): Formattare un testo.

Riempiamo le due sottosezioni del nostro documento. Nella prima sottosezione, inserire il testo seguente:

Questo è il testo della **prima** sottosezione.
Il suo contenuto è di alta caratura *intellettuale*.

Nella seconda sottosezione, inserire il testo seguente:

La seconda **sotto**sezione è ancora più complicata della prima.

Ho lasciato una riga vuota e mezzo per sottolinearlo.

Sotto un'altra riga vuota, creata con `\\[\\baselineskip]`, una nota importante.

Testo, parte 3: dimensioni del font

Si possono cambiare le dimensioni del font in una porzione di testo racchiudendolo tra graffe assieme ad un'apposita macro.

```
{\macroname testo da ridimensionare}
```

Nella frase sopra:

```
{\scriptsize ad un'apposita macro}.
```

Testo, parte 3: dimensioni del font

L^AT_EX mette a disposizione dieci macro di dimensioni di font, definite in base alla dimensione scelta in `\documentclass` (che diventa la `\normalsize`):

<code>\Huge</code>	Text
<code>\huge</code>	Text
<code>\LARGE</code>	Text
<code>\Large</code>	Text
<code>\large</code>	Text
<code>\normalsize</code>	Text
<code>\small</code>	Text
<code>\footnotesize</code>	Text
<code>\scriptsize</code>	Text
<code>\tiny</code>	Text

Elenchi: gli ambienti `itemize` e `enumerate`

Elenchi non numerati:

```
\begin{itemize}  
    \item item 1  
    \item item 2  
\end{itemize}
```

- item 1
- item 2

Elenchi numerati:

```
\begin{enumerate}  
    \item item 1  
    \item item 2  
\end{enumerate}
```

1. item 1
2. item 2

Elenchi: gli ambienti `itemize` e `enumerate`

Si possono cambiare i *bullet point* come argomento opzionale di `\item`.

```
\begin{itemize}
  \item Item 0
  \item[-] Item 1
  \item[(a)] Item 2
  \item[(b)] Item 3
  \item[\textbf{Hi}] Item 4
\end{itemize}
```

- Item 0
- Item 1
- (a) Item 2
- (b) Item 3
- Hi** Item 4

Esercizio 3: Un elenco di grandezze di testo (o la quiete prima della tempesta)

Generare, in un'apposita sezione, il seguente elenco:

1. small,
2. normalsize,
3. large,
4. huge,
5. Huge.

Bonus points!

Includere (nel preambolo) il pacchetto `enumitem` con l'opzione `shortlabels` e dare l'opzione `(i)` alla chiamata dell'ambiente `enumerate` (nel corpo). Cosa accade?

Matematica, parte 1: ambienti di scrittura

Esistono due principali ambienti di scrittura matematica.

- L'ambiente *inline* ha una sintassi a sé:

```
$ ... $
```

- L'ambiente *equation* (a volte chiamato *display*)

```
\begin{equation}  
    contenuto ...  
\end{equation}
```

L'ambiente *inline* serve a includere formule in un testo, come $(a + b)^2 = a^2 + 2ab + b^2$, mentre l'ambiente *equation* fornisce al contenuto uno spazio dedicato nella pagina, numerandolo:

$$(a + b)^2 = a^2 + 2ab + b^2. \quad (1)$$

Matematica, parte 1: ambienti di scrittura

Esistono due principali ambienti di scrittura matematica.

- L'ambiente *inline* ha una sintassi a sé:

```
$ ... $
```

- L'ambiente *equation* (a volte chiamato *display*)

```
\begin{equation}  
    contenuto ...  
\end{equation}
```

L'ambiente *inline* serve a includere formule in un testo, come $(a + b)^2 = a^2 + 2ab + b^2$, mentre l'ambiente *equation* fornisce al contenuto uno spazio dedicato nella pagina, numerandolo:

$$(a + b)^2 = a^2 + 2ab + b^2. \quad (1)$$

Matematica, parte 1: ambienti di scrittura

Esistono due principali ambienti di scrittura matematica.

- L'ambiente *inline* ha una sintassi a sé:

```
$ ... $
```

- L'ambiente *equation* (a volte chiamato *display*)

```
\begin{equation}  
    contenuto ...  
\end{equation}
```

L'ambiente *inline* serve a includere formule in un testo, come $(a + b)^2 = a^2 + 2ab + b^2$, mentre l'ambiente *equation* fornisce al contenuto uno spazio dedicato nella pagina, numerandolo:

$$(a + b)^2 = a^2 + 2ab + b^2. \quad (1)$$

Matematica, parte 1: ambienti di scrittura

L'ambiente `\textit{inline}` serve a includere formule in un testo, come $(a+b)^2 \neq a^2+b^2$, mentre l'ambiente `\textit{equation}` fornisce al contenuto uno spazio dedicato nella pagina, numerandolo:

```
\begin{equation}
      (a+b)^2 \neq a^2+b^2.
\end{equation}
```

L'ambiente *inline* serve a includere formule in un testo, come $(a+b)^2 \neq a^2+b^2$, mentre l'ambiente *equation* fornisce al contenuto uno spazio dedicato nella pagina, numerandolo:

$$(a+b)^2 \neq a^2+b^2. \tag{1}$$

Digressione: numerazione o no?

Regola (quasi) generale

Ogni comando o ambiente che produce una numerazione possiede una versione non numerata, ottenuta aggiungendo al nome del comando (o ambiente) un asterisco.

Ad esempio,

`\subsection{title}` → `\subsection*{title}`

$$\begin{cases} \code{\begin{equation}} & \rightarrow \code{\begin{equation*}} \\ \code{\end{equation}} & \rightarrow \code{\end{equation*}} \end{cases}$$

Warning!

Tutti i comandi presentati di seguito funzionano solo se inseriti in un ambiente di scrittura matematica (qualunque). La sintassi non dipende dall'ambiente scelto.

Matematica, parte 2: simboli e operazioni comuni

123, abc

∞

$a^{\text{above}}, a_{\text{under}}$

$\sqrt[a]{b}$

\sum_a^b

\int_a^b

$\frac{a}{b}$

$\forall, \exists, x \in A$

$>, <, \geq, \leq, \gg, \ll$

\sim, \approx, \neq, \pm

$\subset, \subseteq, \cup, \cap$

\sin, \cos, \tan, \log

123, *abc*

∞

$a^{\text{above}}, a_{\text{under}}$

$\sqrt[a]{b}$

\sum_a^b

\int_a^b

$\frac{a}{b}$

$\forall, \exists, x \in A$

$>, <, \geq, \leq, \gg, \ll$

\sim, \approx, \neq, \pm

$\subset, \subseteq, \cup, \cap$

\sin, \cos, \tan, \log

In *inline*, ad esempio,

```
$ \english_name_of_the_greek_letter $
```

Tre principali varianti:

- lowercase: `\theta` (θ)
- uppercase: `\Theta` (Θ)
- variabili: `\vartheta` (ϑ)

Grassetto e corsivo dovrebbero essere chiamati con questi comandi:

`\mathbf{textBold}` genera **textBold**

`\mathit{textIta}` genera *textBold*

Alcune più specifiche formattazioni usate:

`\mathcal{C}` writes \mathcal{C}

`\mathbb{R}` writes \mathbb{R}

`\mathrm{d}x` writes dx

Grassetto e corsivo dovrebbero essere chiamati con questi comandi:

`\mathbf{textBold}` genera **textBold**

`\mathit{textIta}` genera *textBold*

Alcune più specifiche formattazioni usate:

`\mathcal{C}` writes \mathcal{C}

`\mathbb{R}` writes \mathbb{R}

`\mathrm{d}x` writes $\mathrm{d}x$

Importante!

Negli ambienti matematici, tutti gli spazi nel codice vengono ignorati.

Vedremo più avanti come introdurre spaziature in ambienti matematici.

Esercizio 4: Il Teorema di Lagrange

Sia $[a, b]$ un intervallo reale e sia $f : [a, b] \rightarrow \mathbb{R}$ una funzione continua in $[a, b]$ e derivabile in (a, b) . Esiste $\xi \in (a, b)$ tale che

$$f'(\xi) = \frac{f(b) - f(a)}{b - a}.$$

Esercizio 4: Il Teorema di Lagrange

Sia $[a, b]$ un intervallo reale e sia $f : [a, b] \rightarrow \mathbb{R}$ una funzione continua in $[a, b]$ e derivabile in (a, b) . Esiste $\xi \in (a, b)$ tale che

$$f'(\xi) = \frac{f(b) - f(a)}{b - a}.$$

```
Sia  $[a,b]$  un intervallo reale e sia  $f: [a,b]$   
   $\rightarrow \mathbb{R}$  una funzione continua in  $[a,$   
   $b]$  e derivabile in  $(a,b)$ . Esiste  $\xi \in$   
   $(a,b)$  tale che
```

```
 $\begin{equation*}$   
       $f'(\xi) = \frac{f(b) - f(a)}{b - a}.$   
 $\end{equation*}$ 
```

Dentro all'ambiente `equation` (o `equation*`) possiamo annidare

- l'ambiente `split` per dividere un'espressione su più righe;
- l'ambiente `cases` per sistemi di equazioni.

Matematica, parte 6: i sotto-ambienti split e cases

Nell'ambiente `split`, si va a capo con `\\` e si allineano le righe con `&`:

```
\begin{equation}
  \begin{split}
    (a + b)^2 &= (a+b)(a+b) \\
    &= a^2 + ba + ab + b^2 \\
    &= a^2 + 2ab + b^2.
  \end{split}
\end{equation}
```

$$\begin{aligned}(a + b)^2 &= (a + b)(a + b) \\ &= a^2 + ba + ab + b^2 \\ &= a^2 + 2ab + b^2.\end{aligned}\tag{2}$$

Notate che l'intero blocco (e non ciascuna riga) viene numerato.

Matematica, parte 6: i sotto-ambienti split e cases

Anche nell'ambiente cases si va a capo con `\\` e si allineano le righe con `&`:

```
\begin{equation}
  \begin{cases}
    x + y - 8z & = 0 \\
    x - y + z & = 0 \\
    3x + z & = 0.
  \end{cases}
\end{equation}
```

$$\begin{cases} x + y - 8z & = 0 \\ x - y + z & = 0 \\ 3x + z & = 0. \end{cases} \quad (3)$$

Notare che l'intero blocco (e non ciascuna riga) viene numerato.

Matematica, parte 7: altri ambienti di scrittura matematica

L'ambiente `multline` (e versione non numerata `multline*`) permette di dividere su più righe espressioni troppo lunghe per una sola riga, andando a capo con `\\`.

```
\begin{multline}
    p(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 \\
    + x^8 + x^9 + x^{10} + x^{11} + x^{12}
\end{multline}
```

$$p(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 \\ + x^8 + x^9 + x^{10} + x^{11} + x^{12} \quad (4)$$

La parte "in eccesso" viene automaticamente allineata a destra.
Usare `multline` solo se ce n'è bisogno!

Matematica, parte 7: altri ambienti di scrittura matematica

L'ambiente `alignat` (e versione non numerata `alignat*`) permette di scrivere espressioni su più righe, allineandole con `\\` e `&`. Richiede il numero di righe.

```
\begin{alignat}{3}
  f & \& \text{continua in ogni punto di } [3,4]; \\
  g & \& \text{continua in ogni punto di } [3,4]; \\
  f+g & \& \text{continua in ogni punto di } [3,4].
\end{alignat}
```

$$f \text{ continua in ogni punto di } [3,4]; \quad (5)$$

$$g \text{ continua in ogni punto di } [3,4]; \quad (6)$$

$$f + g \text{ continua in ogni punto di } [3,4]. \quad (7)$$

Rispetto a `cases` e `split` è possibile utilizzare più `&` per riga. Notate che ogni equazione viene numerata.

L'ambiente `align` (e versione non numerata `align*`) fa la stessa cosa, assumendo di allineare equazioni.

```
\begin{align*}
    x_1 &= x_2 & x_3 &= x_4 \\\
    x_5 &= x_6 & x_7 &= x_8.
\end{align*}
```

$$x_1 = x_2,$$

$$x_3 = x_4,$$

$$x_5 = x_6,$$

$$x_7 = x_8.$$

Le spaziature non sono uniformi.

Matematica, parte 8: delimitatori

I delimitatori sono parentesi o oggetti che si comportano come parentesi, usati per racchiudere un'espressione (e.g. (\cdot) , $[\cdot]$, $\{\cdot\}$, $|\cdot|$, $\|\cdot\|$).

Se usati, devono essere aperti e chiusi, oltre che correttamente annidati.

Importante!

È buona norma che i delimitatori abbiano la grandezza di ciò che contengono. Usate però anche il vostro senso estetico!

Matematica, parte 8: delimitatori

I delimitatori sono parentesi o oggetti che si comportano come parentesi, usati per racchiudere un'espressione (e.g. (\cdot) , $[\cdot]$, $\{\cdot\}$, $|\cdot|$, $\|\cdot\|$).

Se usati, devono essere aperti e chiusi, oltre che correttamente annidati.

Importante!

È buona norma che i delimitatori abbiano la grandezza di ciò che contengono. Usate però anche il vostro senso estetico!

I delimitatori sono parentesi o oggetti che si comportano come parentesi, usati per racchiudere un'espressione (e.g. (\cdot) , $[\cdot]$, $\{\cdot\}$, $|\cdot|$, $\|\cdot\|$).

Se usati, devono essere aperti e chiusi, oltre che correttamente annidati.

Importante!

È buona norma che i delimitatori abbiano la grandezza di ciò che contengono. Usate però anche il vostro senso estetico!

Matematica, parte 8: delimitatori

I comandi `\left` e `\right` seguiti dal delimitatore scelto, adattano dinamicamente le grandezze dei delimitatori al contenuto.

```
\begin{equation*}
    (1 + \frac{1}{n})^n,
    \left( 1 + \frac{1}{n} \right)^n.
\end{equation*}
```

$$\left(1 + \frac{1}{n}\right)^n, \quad \left(1 + \frac{1}{n}\right)^n.$$

Ad ogni `\left` deve corrispondere un `\right`. Parentesi tonde, quadre e stanghette singole possono essere scritte da tastiera. Le graffe hanno bisogno di escaping: `\left\{`, `\right\}`, mentre le stanghette doppie si generano con `\left|`, `\right|`.

Matematica, parte 8: delimitatori

I comandi `\left` e `\right` seguiti dal delimitatore scelto, adattano dinamicamente le grandezze dei delimitatori al contenuto.

```
\begin{equation*}
    (1 + \frac{1}{n})^n,
    \left( 1 + \frac{1}{n} \right)^n.
\end{equation*}
```

$$\left(1 + \frac{1}{n}\right)^n, \quad \left(1 + \frac{1}{n}\right)^n.$$

Ad ogni `\left` **deve** corrispondere un `\right`. Parentesi tonde, quadre e stanghette singole possono essere scritte da tastiera. Le graffe hanno bisogno di escaping: `\left\{`, `\right\}`, mentre le stanghette doppie si generano con `\left|`, `\right|`.

Matematica, parte 9: matrici

Esistono ambienti dedicati per le matrici.

```
\begin{equation*}
  \begin{bmatrix}
    a & b \\
    c & d
  \end{bmatrix}
\end{equation*}
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Varianti sul tema:

· `matrix` senza delimitatore

$$\begin{matrix} a & b \\ c & d \end{matrix}$$

· `pmatrix` parentesi tonde

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

· `vmatrix` stanghette singole

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

Matematica, parte 9: matrici

Esistono ambienti dedicati per le matrici.

```
\begin{equation*}
  \begin{bmatrix}
    a & b \\
    c & d
  \end{bmatrix}
\end{equation*}
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Varianti sul tema:

- `matrix` senza delimitatore $\begin{matrix} a & b \\ c & d \end{matrix}$
- `pmatrix` parentesi tonde $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$
- `vmatrix` stanghette singole $\begin{vmatrix} a & b \\ c & d \end{vmatrix}$

Matematica, parte 10: spacing

Alcune lunghezze macro utili per spaziare in ambiente matematico (e non):

<code>\quad</code>	circa 4 spazi	$x_1 \quad x_2$
<code>\qquad</code>	circa 8 spazi	$x_1 \qquad x_2$
<code>\,</code>	circa 1 spazio	$x_1 x_2$
<code>\:</code>	circa 1 spazio	$x_1 x_2$
<code>\!</code>	circa -1 spazio	$x_1 x_2$
<code>\</code>	esattamente 1 spazio	$x_1 x_2$

Esempio di utilizzo:

```
$x_1 \quad \textcolor{blue}{\qquad} x_2$
```


Esercizio 5: Equazioni!

Replicare le seguenti equazioni (è sufficiente elencarle, ignorare il formato 2x2):

$$\sin^2(x) + \cos^2(x) = 1$$

$$\begin{cases} (x+y)^{\alpha\beta+\delta} &= 1, \\ \cos(xy) &= 0. \end{cases}$$

$$\frac{d^2f}{dx^2} + \frac{df}{dx} = 4f - 1$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Bonus points!

Con i comandi `\sup` e `\inf`, che ammettono come pedice la loro specifica, replicare anche

$$\sup_{x \in [0,1]} \left(\inf_{y \in [0,1]} |\mathbf{g}(x,y)| - \sqrt{|\mathbf{g}(x,y)|} \right) = \frac{3}{4}.$$

Grazie dell'attenzione!

Prossima lezione: 2/12/2021 in 3.1.6