

Laboratorio di Introduzione alla Matematica
Computazionale – A.A. 2024/2025
02 - L^AT_EX

Fabio Durastante <fabio.durastante@unipi.it>,
Silvia Gazzola <silvia.gazzola@unipi.it>

– Basato sulle slide dell'A.A. 2020/21 di L. Robol e S. Steffè –

3/4 Ottobre 2023

T_EX e L^AT_EX

- ▶ Donald Knuth è l'autore di "The Art of Computer programming".
- ▶ Quando, alla fine degli anni 70, ottenne le bozze della seconda ristampa, rimase inorridito dalla bassa qualità tipografica.
- ▶ ... e decise che doveva esistere qualche metodo più intelligente di affrontare il problema.
- ▶ Così è nato il T_EX.



T_EX e L^AT_EX

- ▶ Donald Knuth è l'autore di “The Art of Computer programming”.
- ▶ Quando, alla fine degli anni 70, ottenne le bozze della seconda ristampa, rimase inorridito dalla bassa qualità tipografica.
- ▶ ... e decise che doveva esistere qualche metodo più intelligente di affrontare il problema.
- ▶ Così è nato il T_EX.



Ogni matematico ad un certo punto incontra il T_EX – e certamente toccherà anche a voi quando dovrete scrivere la tesi oppure presentare un seminario.

Un po' di storia

- ▶ Knuth cominciò a lavorare al T_EX nel 1977, progettando di finirlo nel seguente anno.

Preliminary preliminary description of TEX
D Knuth, May 13, 1977

In this memo I will try to explain the TEX system for preparing publishable documents and how it is proposed to implement the system. Even though I don't understand TEX very well myself yet, I think the best way for me to get started is by trying to explain it to somebody else.

TEX is for technical text. Insiders pronounce the X as a Greek Chi (cf. the Scottish 'ch' sound in 'Loch Ness') since the word 'technical' stems from a Greek root meaning art as well as technology. I am preparing the system primarily for use in publishing my series The Art of Computer Programming— the initial system will be tuned for my books, but it will not be difficult to extend it for other purposes if anybody wants to do so.

Un po' di storia

- ▶ Knuth cominciò a lavorare al $\text{T}_{\text{E}}\text{X}$ nel 1977, progettando di finirlo nel seguente anno.
- ▶ In realtà, la versione finale fu rilasciata solo nel 1989 ($\text{T}_{\text{E}}\text{X}$ 3); da quel momento le nuove versioni hanno un numero che converge a π .
- ▶ Ad esempio, l'ultima release del $\text{T}_{\text{E}}\text{X}$ è la versione 3.14159265. Il numero di versione verrà settato a π — e lo sviluppo congelato — dopo la morte di Knuth.
- ▶ In maniera simile, il sistema di compilazione dei caratteri usati dal sistema $\text{T}_{\text{E}}\text{X}$ (MetaFont) ha raggiunto la versione 2, e da allora converge a $e = 2.71828\dots$

Un po' di storia

- ▶ Knuth cominciò a lavorare al T_EX nel 1977, progettando di finirlo nel seguente anno.
- ▶ In realtà, la versione finale fu rilasciata solo nel 1989 (T_EX 3); da quel momento le nuove versioni hanno un numero che converge a π .
- ▶ Knuth offre **premi in denaro** alle persone che **trovano** e **segnalano** un **bug** in T_EX. Il premio per bug è iniziato a 2,56 \$ e raddoppiato ogni anno fino a quando non è stato congelato al suo valore attuale di 327,68 \$.

Un po' di storia

- ▶ Knuth cominciò a lavorare al T_EX nel 1977, progettando di finirlo nel seguente anno.
- ▶ In realtà, la versione finale fu rilasciata solo nel 1989 (T_EX 3); da quel momento le nuove versioni hanno un numero che converge a π .
- ▶ Knuth offre **premi in denaro** alle persone che **trovano** e **segnalano** un **bug** in T_EX. Il premio per bug è iniziato a 2,56 \$ e raddoppiato ogni anno fino a quando non è stato congelato al suo valore attuale di 327,68 \$.
- ▶ Knuth ha perso relativamente pochi soldi in quanto sono stati rivendicati pochissimi bug. . . e i destinatari incorniciano il loro assegno come prova di aver trovato un bug in T_EX piuttosto che incassarlo. . .

Funzionamento del T_EX

Il T_EX è un software di typesetting – ma anche un linguaggio di programmazione; dato un file in linguaggio T_EX, compilarlo con il comando `tex` genera in output un file PDF.

- ▶ L'interprete T_EX legge un file `esempio.tex`, tramite il comando `pdftex`.
- ▶ Valuta i vari comandi, e produce dei file ausiliari, insieme ad un file `esempio.pdf`.

Esempio minimale

Proviamo a compilare un documento contenente

Definiamo $y = x^2 - ab$.

`\bye`

e otteniamo

$$\textit{Definiamo } y = x^2 - ab$$

Esempio minimale

Proviamo a compilare un documento contenente

Definiamo $y = x^2 - ab$.

`\bye`

e otteniamo

$$Definiamo y = x^2 - ab$$

Oppure possiamo usare **comandi** T_EX, che cominciano con il simbolo `\`:

```
\[  
\frac{\partial u}{\partial t} =  
\int_0^1 u(x,t) \varphi(x) dx  
\]
```

che produce

$$\frac{\partial u(x, t)}{\partial t} = \int_0^1 u(x, t) \varphi(x) dx$$

Macro

- ▶ In T_EX è possibile definire delle **macro**, ovvero “comandi aggiuntivi”.
- ▶ Queste si possono usare per estendere il T_EX – e fornire funzionalità aggiuntive. Tipicamente queste funzionalità vengono raggruppate in **pacchetti**.
- ▶ Ad esempio, c'è un pacchetto per le lettere gotiche, un pacchetto per scrivere del codice, un pacchetto per le slide (Beamer, che ho usato per questi lucidi), un pacchetto per disegnare, per fare grafici, ecc.
Un elenco di pacchetti disponibili si trova su:
The Comprehensive T_EX Archive Network
<https://www.ctan.org/>
- ▶ Il T_EX è un linguaggio di programmazione, e come spesso succede andare a capo non è troppo diverso da inserire uno spazio; per andare a capo nel testo bisogna usare `\\`, o lasciare una riga vuota.

- ▶ Nell'uso di tutti i giorni, è necessario avere un set di regole preconfezionate per il layout di pagina – altrimenti è facile incorrere in errori tipografici.
- ▶ Per questa ragione, è stato sviluppato il L^AT_EX, un set di **macro** T_EX che permette all'utente di descrivere la struttura del testo, senza preoccuparsi (troppo) della formattazione.
- ▶ Il L^AT_EX fornisce degli “ambienti”.
- ▶ Impareremo anche questo in laboratorio.

Editor per L^AT_EX

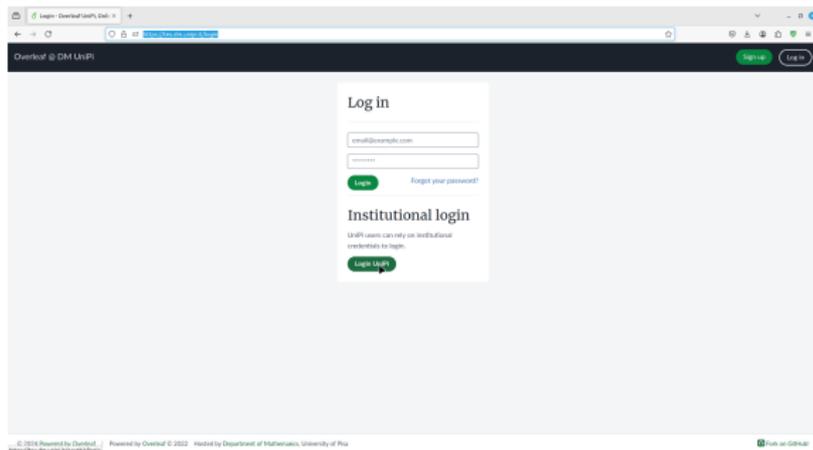
Si può utilizzare **qualsunque editor di testo** che vi permetta di salvare dei file *plain text* con estensione `.tex` in combinazione ad una *shell* su cui eseguire il comando `pdflatex`.

Ci sono tuttavia degli **editor specifici** che permettono di utilizzare alcuni **automatismi**, e.g.,

- ▶ **Overleaf**: Editor online collaborativo con anteprima in tempo reale. Non richiede installazione.
- ▶ **TeXstudio**: Editor desktop gratuito con supporto per la compilazione e suggerimenti di codice.
- ▶ **TeXmaker**: Editor leggero e multi-piattaforma con una GUI semplice.
- ▶ **Kile**: Editor L^AT_EX per sistemi Linux, con molte funzionalità avanzate.
- ▶ **LyX**: Un editor WYSIWYG che nasconde la complessità di L^AT_EX per un approccio visuale.

Overleaf di dipartimento

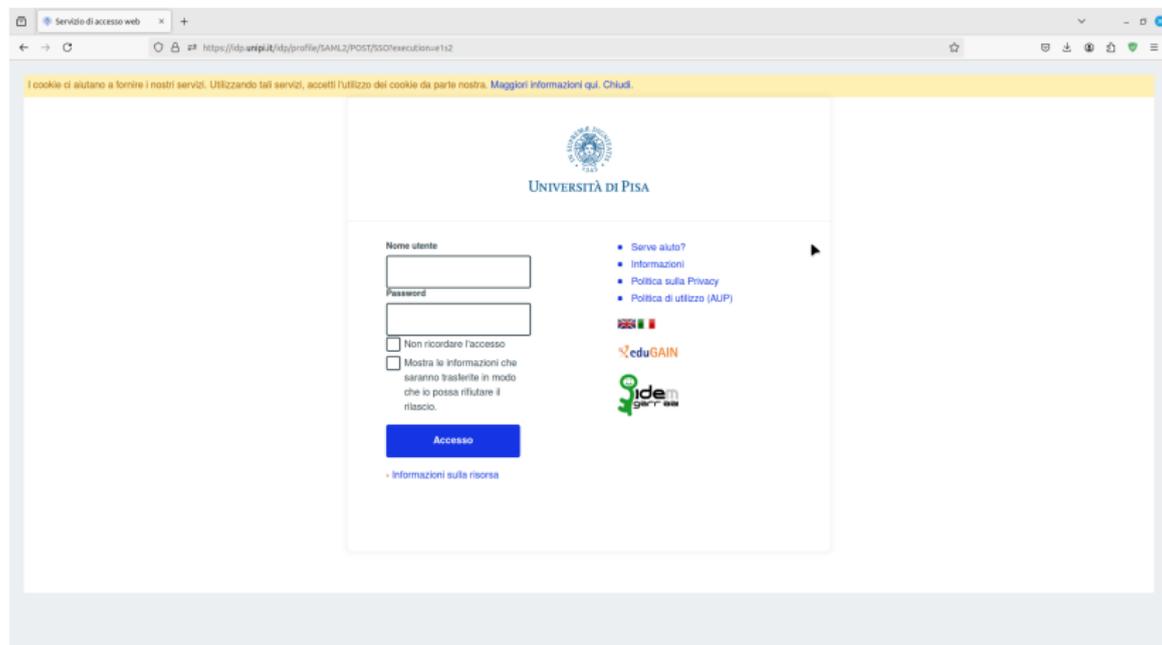
Abbiamo installato su una **macchina virtuale** nostra una versione online del software OpenSource Overleaf a cui potete **accedere** con le vostre **credenziali di ateneo**:



`https://tex.dm.unipi.it/login`

Il bottone **Log in UniPi** vi porta alla solita pagina di login per le credenziali di ateneo.

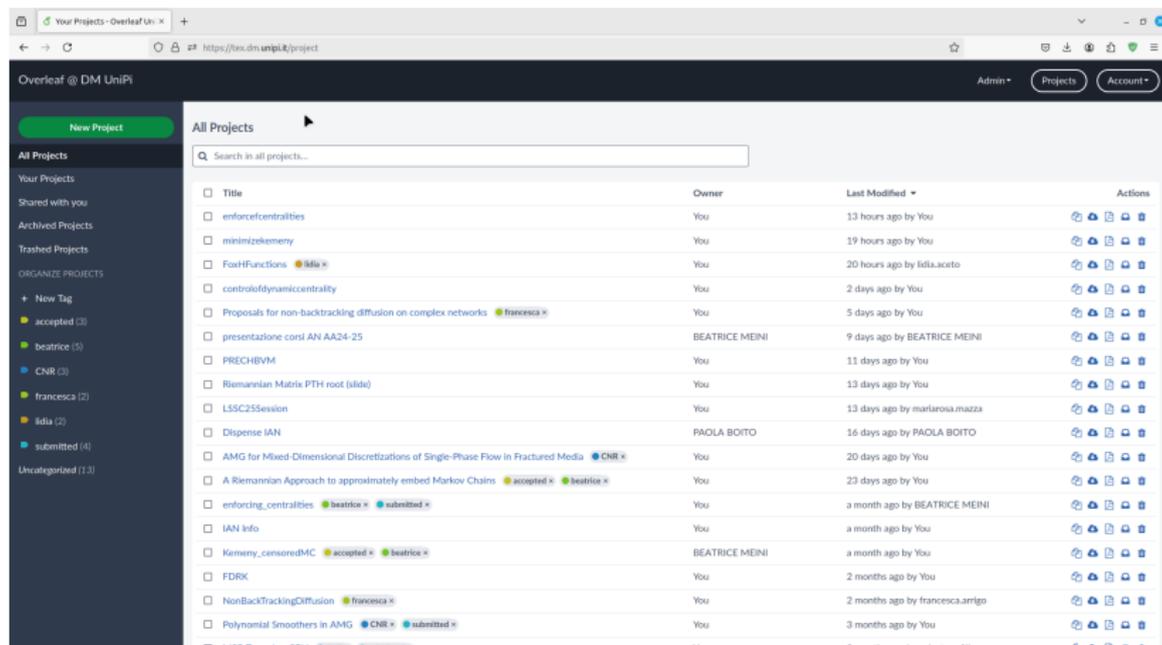
Overleaf di dipartimento



The screenshot shows a web browser window with the following elements:

- Browser Tab:** Servizio di accesso web
- Address Bar:** <https://idp.unipi.it/Idp/profile/SAML2/POST/SSO?execution=112>
- Cookie Notice:** I cookie ci aiutano a fornire i nostri servizi. Utilizzando tali servizi, accetti l'utilizzo dei cookie da parte nostra. [Maggiori informazioni qui.](#) [Chiudi.](#)
- University Logo:** UNIVERSITÀ DI PISA
- Form Fields:**
 - Nome utente:
 - Password:
- Options:**
 - Non ricordare l'accesso
 - Mostra le informazioni che saranno trasferite in modo che io possa rifiutare il rilascio.
- Buttons:**
 - [Accesso](#) (blue button)
 - [Informazioni sulla risorsa](#) (link)
- Help Links:**
 - [Serve aiuto?](#)
 - [Informazioni](#)
 - [Politica sulla Privacy](#)
 - [Politica di utilizzo \(AUP\)](#)
- Logos:** Italian flag, eduGAIN, and iDeM logos.

Overleaf di dipartimento



The screenshot shows the Overleaf web interface for a department. At the top, there's a navigation bar with 'Admin', 'Projects', and 'Account' buttons. Below that, the main content area is titled 'All Projects'. On the left, there's a sidebar with a 'New Project' button and a list of project categories: 'Your Projects', 'Archived Projects', 'Trashed Projects', and 'ORGANIZE PROJECTS'. The 'ORGANIZE PROJECTS' section includes a 'New Tag' button and several tags like 'accepted', 'beatrice', 'CNR', 'francesca', 'lidia', and 'submitted'. The main area contains a search bar and a table of projects.

Title	Owner	Last Modified	Actions
<input type="checkbox"/> enforcecentralities	You	13 hours ago by You	
<input type="checkbox"/> minimizsekemery	You	19 hours ago by You	
<input type="checkbox"/> FooHFunctions lidia x	You	20 hours ago by lidia.aceto	
<input type="checkbox"/> controllodynamiccentralty	You	2 days ago by You	
<input type="checkbox"/> Proposals for non-backtracking diffusion on complex networks francesca x	You	5 days ago by You	
<input type="checkbox"/> presentazione corsi AN AA24-25	BEATRICE MEINI	9 days ago by BEATRICE MEINI	
<input type="checkbox"/> PRECHBVM	You	11 days ago by You	
<input type="checkbox"/> Riemannian Matrix PTH root (lidia)	You	13 days ago by You	
<input type="checkbox"/> L55C2Session	You	13 days ago by marlarosa.mazza	
<input type="checkbox"/> Dispense IAN	PAOLA BOITO	16 days ago by PAOLA BOITO	
<input type="checkbox"/> AMG for Mixed-Dimensional Discretizations of Single-Phase Flow in Fractured Media CNR x	You	20 days ago by You	
<input type="checkbox"/> A Riemannian Approach to approximately embed Markov Chains accepted x beatrice x	You	23 days ago by You	
<input type="checkbox"/> enforcing_centralities beatrice x submitted x	You	a month ago by BEATRICE MEINI	
<input type="checkbox"/> IAN info	You	a month ago by You	
<input type="checkbox"/> Kermery_censoredMC accepted x beatrice x	BEATRICE MEINI	a month ago by You	
<input type="checkbox"/> FDRK	You	2 months ago by You	
<input type="checkbox"/> NonBackTrackingDiffusion francesca x	You	2 months ago by francesca.amigo	
<input type="checkbox"/> Polynomial Smoothers in AMG CNR x submitted x	You	3 months ago by You	

In cui potete creare un nuovo progetto dal bottone “New Project” in alto a sinistra.

Documento L^AT_EX di esempio

```
\documentclass{article}
```

```
\title{Il mio documento}
```

```
\author{Johann Friedrich Carl Gau\ss}
```

```
\begin{document}
```

```
\maketitle
```

```
\section{Introduzione}
```

Questo documento presenta la principali caratteristiche del \LaTeX ; possiamo scrivere matematica in linea

($a = 2 \cdot b$), o in equazioni

```
\begin{equation}
```

```
\frac{1}{2\pi i} \int_{\partial \Gamma} f(z) dz = 0.
```

```
\end{equation}
```

```
\end{document}
```

Cos'è una classe in \LaTeX

Una classe in \LaTeX definisce la struttura e lo stile generale di un documento. Alcune caratteristiche principali delle classi:

- ▶ **Definizione:** Una classe in \LaTeX è un insieme di regole che determina l'aspetto e il comportamento del documento (layout, dimensione del testo, numerazione, ecc.).
- ▶ **Sintassi:** Il comando `\documentclass{classe}` viene utilizzato per specificare la classe del documento.
- ▶ **Esempi di classi:**
 - ▶ `article`: per articoli brevi.
 - ▶ `report`: per documenti più lunghi, come rapporti o tesi.
 - ▶ `book`: per libri.
 - ▶ `beamer`: per presentazioni.
- ▶ **Personalizzazione:** È possibile creare o modificare classi personalizzate per specifici tipi di documenti.

Preambolo e Documento in L^AT_EX

Un file L^AT_EX è diviso in due parti principali: il **preambolo** e il **documento**.

▶ **Preambolo:**

- ▶ Inizia dopo `\documentclass{}`.
- ▶ Contiene configurazioni generali come:
 - ▶ Importazione di pacchetti (`\usepackage`).
 - ▶ Definizione di comandi personalizzati.
 - ▶ Impostazioni del layout (margini, font, ecc.).
- ▶ Non viene visualizzato nel documento finale.

▶ **Documento:**

- ▶ Inizia con `\begin{document}` e termina con `\end{document}`.
- ▶ Contiene il contenuto effettivo del documento: testo, figure, tabelle, ecc.
- ▶ È la parte visibile nel risultato finale.

Font e Dimensioni in \LaTeX

In \LaTeX , i font e le loro dimensioni possono essere gestiti in modo flessibile. Ecco alcuni punti chiave:

- ▶ **Font Predefiniti:**

- ▶ Computer Modern: il font predefinito di \LaTeX .
- ▶ Altri font comuni includono Latin Modern, Times, e Palatino.

- ▶ **Dimensioni dei Font:**

- ▶ Dimensioni standard:
 - ▶ 10pt: dimensione di default.
 - ▶ 11pt e 12pt: opzioni comuni per documenti più leggibili.
- ▶ Esempi di comandi per modificare la dimensione:
 - ▶ `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\HUGE`.

- ▶ **Esempio di codice:**

```
\textbf{Questo è in grassetto.}
{\Large Questo è grande.}
{\footnotesize Questo è piccolo.}
```

Ambienti

La differenza essenziale fra $\text{T}_{\text{E}}\text{X}$ e \LaTeX è la presenza di **ambienti**, che si aprono con il comando `\begin` e si chiudono con `\end`:

- ▶ `\begin{document} ... \end{document}` contiene tutto il testo.
- ▶ `\begin{equation} ... \end{equation}` contiene un'equazione numerata.
- ▶ ...

Ambienti: qualche esempio di dettaglio

- ▶ L'ambiente viene avviato con `\begin{itemize}`.
- ▶ Ogni elemento viene introdotto con il comando `\item`.
- ▶ L'ambiente si chiude con `\end{itemize}`.
- ▶ I punti elenco vengono automaticamente numerati con pallini.
- ▶ È possibile nidificare gli elenchi.

Esempio di codice:

Esempio

```
\begin{itemize}
\item Primo elemento
\item Secondo elemento
\item Terzo elemento
\end{itemize}
```

Ambienti: qualche esempio di dettaglio

L'ambiente `enumerate` viene utilizzato per creare elenchi numerati in \LaTeX . Di seguito alcune caratteristiche:

1. L'ambiente inizia con `\begin{enumerate}`.
2. Ogni elemento viene introdotto con `\item`.
3. La numerazione è automatica e può essere personalizzata.
4. L'ambiente si chiude con `\end{enumerate}`.
5. Gli elenchi numerati possono essere nidificati.

Esempio di codice:

Esempio

```
\begin{enumerate}
\item Primo elemento
\item Secondo elemento
\item Terzo elemento
\end{enumerate}
```

Ambienti: qualche esempio di dettaglio

1. Enumerate

▶ Itemize

```
\begin{enumerate}  
\item Enumerate  
\begin{itemize}  
\item Itemize  
\end{itemize}  
\end{enumerate}
```

▶ Itemize

1. Enumerate

```
\begin{enumerate}  
\item Itemize  
\begin{itemize}  
\item Enumerate  
\end{itemize}  
\end{enumerate}
```

1. Enumerate

1.1 Enumerate

```
\begin{enumerate}  
\item Enumerate  
\begin{enumerate}  
\item Enumerate  
\end{enumerate}  
\end{enumerate}
```

Potete **provare allo stesso modo** ad innestare più di due ambienti di tipo `itemize` o `enumerate`.

Ambienti: qualche esempio di dettaglio

L'ambiente `description` viene utilizzato per creare elenchi descrittivi, in cui ogni voce ha una parola chiave che viene evidenziata. Caratteristiche principali:

Sintassi: L'ambiente si apre con `\begin{description}` e si chiude con `\end{description}`.

Chiave: Ogni elemento ha una "chiave" tra parentesi quadre che descrive brevemente l'elemento.

Uso comune: È ideale per definizioni o elenchi dove si vogliono evidenziare concetti.

Esempio di codice:

Esempio

```
\begin{description}
\item[LaTeX:] Un linguaggio di markup per la tipografia.
\item[Beamer:] Un pacchetto per presentazioni in LaTeX.
\item[Description:] Un ambiente per elenchi descrittivi.
\end{description}
```

Gli ambienti `quote` e `quotation` in \LaTeX

In \LaTeX , gli ambienti `quote` e `quotation` vengono utilizzati per formattare citazioni di testo. Ecco le differenze principali:

- ▶ `quote`:
 - ▶ Utilizzato per brevi citazioni.
 - ▶ Il testo è indentato ma non ha interruzioni tra i paragrafi.
- ▶ `quotation`:
 - ▶ Ideale per citazioni più lunghe e con più paragrafi.
 - ▶ Oltre all'indentazione, separa i paragrafi interni.

Esempi di utilizzo:

Esempio di `quote`

```
\begin{quote}
```

```
Questo è un esempio di una breve citazione.
```

```
\end{quote}
```

Gli ambienti `quote` e `quotation` in \LaTeX

In \LaTeX , gli ambienti `quote` e `quotation` vengono utilizzati per formattare citazioni di testo. Ecco le differenze principali:

- ▶ `quote`:
 - ▶ Utilizzato per brevi citazioni.
 - ▶ Il testo è indentato ma non ha interruzioni tra i paragrafi.
- ▶ `quotation`:
 - ▶ Ideale per citazioni più lunghe e con più paragrafi.
 - ▶ Oltre all'indentazione, separa i paragrafi interni.

Esempi di utilizzo:

Esempio di `quotation`

```
\begin{quotation}
```

Questa è una citazione più lunga, composta da più paragrafi.

Ogni paragrafo è separato dall'altro.

```
\end{quotation}
```

Pacchetti AMS per la matematica in \LaTeX

I pacchetti AMS (*American Mathematical Society*) offrono strumenti avanzati per la scrittura di formule e strutture matematiche complesse in \LaTeX . I principali pacchetti AMS sono:

- ▶ `amsmath`: Migliora la gestione delle equazioni.
- ▶ `amssymb`: Fornisce simboli matematici aggiuntivi.
- ▶ `amsfonts`: Include font aggiuntivi per la matematica.
- ▶ `amsthm`: Gestione di ambienti per Teoremi, Lemmi, *etc.*

Funzionalità principali:

- ▶ `align`: Allinea equazioni su più righe.
- ▶ `gather`: Centra equazioni su più righe.
- ▶ `cases`: Crea sistemi di equazioni con parentesi graffe.
- ▶ Nuovi simboli: `\mathbb`, `\mathfrak`, ecc.

Si inserisce nel preambolo:

Caricare un pacchetto

```
\usepackage{amsmath, amssymb, amsfonts, amsthm}
```

Uso del comando align

Il comando align permette di allineare equazioni su più righe.

Esempio di codice

```
\begin{align}
E &= mc^2 \\
a^2 + b^2 &= c^2
\end{align}
```

Risultato:

$$E = mc^2 \tag{1}$$

$$a^2 + b^2 = c^2 \tag{2}$$

Uso del comando gather

Il comando `gather` centra equazioni su più righe, senza allinearle.

Esempio di codice

```
\begin{gather}
x = y + z \\
x^2 + y^2 = z^2
\end{gather}
```

Risultato:

$$x = y + z \tag{3}$$

$$x^2 + y^2 = z^2 \tag{4}$$

Uso del comando cases

Il comando cases crea sistemi di equazioni o definizioni a pezzi.

Esempio di codice

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

Risultato:

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

Simboli speciali da `amssymb`

Il pacchetto `amssymb` offre simboli matematici aggiuntivi. Alcuni esempi:

- ▶ `\mathbb`: \mathbb{R} , \mathbb{Z} , \mathbb{N}
- ▶ `\forall`: \forall
- ▶ `\exists`: \exists
- ▶ `\neg`: \neg
- ▶ `\infty`: ∞

Esempio di codice:

```
\mathbb{R}, \forall x \in \mathbb{R}, \exists y \infty
```

Risultato:

$$\mathbb{R}, \forall x \in \mathbb{R}, \exists y \infty$$

Uso dei font matematici con amsfonts

Il pacchetto `amsfonts` permette di utilizzare font matematici come `mathbb` e `mathfrak`.

Esempio di codice

```
\mathbb{N}, \mathbb{Z}, \mathfrak{g}, \mathcal{L}
```

Risultato:

$\mathbb{N}, \mathbb{Z}, \mathfrak{g}, \mathcal{L}$

Introduzione al pacchetto `amsthm`

Il pacchetto `amsthm` fornisce strumenti per definire teoremi, definizioni, corollari e lemmi in modo strutturato e uniforme.

- ▶ Offre una gestione semplificata delle dichiarazioni matematiche.
- ▶ Permette di personalizzare le intestazioni di teoremi e definizioni.

Per utilizzare `amsthm`, basta importarlo nel preambolo del documento:

Esempio di codice

```
\usepackage{amsthm}
```

Utilizzo del comando `theoremstyle`

Il comando `\theoremstyle` permette di personalizzare lo stile delle dichiarazioni matematiche.

- ▶ `plain`: Stile predefinito (grassetto, numerato).
- ▶ `definition`: Stile per definizioni (italico, non numerato).
- ▶ `remark`: Stile per osservazioni (italico, non numerato).

Esempio di utilizzo di `theoremstyle`

Ecco come utilizzare `\theoremstyle` per personalizzare i teoremi e le definizioni:

Esempio di codice

```
\theoremstyle{definition}
\newtheorem{definizione}{Definizione}
\theoremstyle{plain}
\newtheorem{teorema}{Teorema}
```

Utilizzo:

```
\begin{definizione}
```

Un numero primo è un numero maggiore di 1 che ha solo due divisori: 1 e se stesso.

```
\end{definizione}
```

```
\begin{teorema}
```

La somma degli angoli interni di un triangolo è 180 gradi.

```
\end{teorema}
```

Definizione di un teorema

Per definire un teorema, utilizziamo il comando `\newtheorem`.

Esempio di codice

```
\newtheorem{teorema}{Teorema}
```

Utilizzo:

```
\begin{teorema}
```

```
Ogni numero pari è divisibile per 2.
```

```
\end{teorema}
```

Definizione di più teoremi

Possiamo definire più tipi di teoremi in un documento:

Esempio di codice

```
\newtheorem{definizione}{Definizione}  
\newtheorem{corollario}[teorema]{Corollario}  
\newtheorem{lemma}[teorema]{Lemma}
```

Utilizzo:

```
\begin{definizione}  
Un quadrato è un poligono con quattro lati di  
uguale lunghezza.  
\end{definizione}
```

Personalizzazione delle intestazioni

Possiamo personalizzare le intestazioni dei teoremi con un'opzione nel comando `\newtheorem`.

Esempio di codice

```
\newtheorem{teorema}{Teorema}[section]
```

Con questa impostazione, i numeri dei teoremi saranno numerati in base alla sezione.

Utilizzo:

```
\begin{teorema}
```

```
Ogni triangolo ha almeno un angolo acuto.
```

```
\end{teorema}
```

Automatismi

\LaTeX si occupa automaticamente di molte cose:

- ▶ La gestione delle sezioni, e i riferimenti; aggiusta anche la spaziatura come necessario secondo le regole tipografiche.
- ▶ La numerazione di equazioni, teoremi, ecc.
- ▶ La gestione della bibliografia.
- ▶ Il posizionamento ottimale di figure e tabelle.
- ▶ Liste puntate, numerate, ecc.

Possiamo usare i comandi `\label` e `\ref` per marcare delle sezioni / teoremi / equazioni, e per creare dei riferimenti.

Comandi di Sezionamento in L^AT_EX

In L^AT_EX, i comandi di sezionamento permettono di organizzare il contenuto del documento in modo chiaro e gerarchico. I principali comandi sono:

- ▶ `\section{Titolo}`: Definisce una sezione principale.
- ▶ `\subsection{Titolo}`: Definisce una sottosezione di una sezione.
- ▶ `\subsubsection{Titolo}`: Definisce una sotto-sottosezione.
- ▶ `\paragraph{Titolo}`: Introduce un paragrafo con intestazione.
- ▶ `\subparagraph{Titolo}`: Introduce un sotto-paragrafo.

Esempio di codice

```
\section{Introduzione}  
\subsection{Contesto}  
\subsubsection{Dettagli}  
\paragraph{Informazioni aggiuntive}  
\subparagraph{Ulteriori dettagli}
```

Comandi di Sezionamento in \LaTeX

In \LaTeX , i comandi di sezionamento permettono di organizzare il contenuto del documento in modo chiaro e gerarchico. I principali comandi sono:

- ▶ `\section{Titolo}`: Definisce una sezione principale.
- ▶ `\subsection{Titolo}`: Definisce una sottosezione di una sezione.
- ▶ `\subsubsection{Titolo}`: Definisce una sotto-sottosezione.
- ▶ `\paragraph{Titolo}`: Introduce un paragrafo con intestazione.
- ▶ `\subparagraph{Titolo}`: Introduce un sotto-paragrafo.
- ▶ Ogni comando genera automaticamente un numero e crea un'entrata nel sommario (se presente).
- ▶ Le intestazioni possono essere personalizzate tramite pacchetti come `titlesec`.

Scrittura di Tabelle in \LaTeX

Le tabelle in \LaTeX possono essere create utilizzando l'ambiente `tabular`. Ecco i punti chiave:

- ▶ **Struttura di base:**

```
\begin{tabular}{opzioni}  
  intestazione 1 & intestazione 2 \\  
  riga 1 col 1 & riga 1 col 2 \\  
  riga 2 col 1 & riga 2 col 2 \\  
\end{tabular}
```

- ▶ **Opzioni delle colonne:**

- ▶ l: Allineamento a sinistra
- ▶ c: Allineamento al centro
- ▶ r: Allineamento a destra

Scrittura di Tabelle in L^AT_EX

► Esempio di tabella:

Esempio di codice

```
\begin{tabular}{|c|c|}  
\hline  
Intestazione 1 & Intestazione 2 \\  
\hline  
Riga 1 Col 1 & Riga 1 Col 2 \\  
\hline  
Riga 2 Col 1 & Riga 2 Col 2 \\  
\hline  
\end{tabular}
```

► Aggiunta di linee:

- `\hline`: Aggiunge una linea orizzontale.
- `\vline`: Aggiunge una linea verticale.

Scrittura di Tabelle in \LaTeX

▶ **Esempio di tabella:**

Intestazione 1	Intestazione 2
Riga 1 Col 1	Riga 1 Col 2
Riga 2 Col 1	Riga 2 Col 2

▶ **Aggiunta di linee:**

- ▶ `\hline`: Aggiunge una linea orizzontale.
- ▶ `\vline`: Aggiunge una linea verticale.

Inserimento di Immagini in \LaTeX

Per inserire immagini in un documento \LaTeX , è necessario utilizzare il pacchetto `graphicx`. Ecco i passaggi principali:

- ▶ **Importazione del pacchetto:**

Esempio di codice

```
\usepackage{graphicx}
```

- ▶ **Comando per inserire un'immagine:**

Esempio di codice

```
\begin{figure}[h]
```

```
\centering
```

```
\includegraphics[width=0.5\textwidth]{nome_immagine.jpg}
```

```
\caption{Descrizione dell'immagine}
```

```
\end{figure}
```

Inserimento di Immagini in L^AT_EX

- ▶ **Opzioni per `\includegraphics`:**
 - ▶ `width`: Imposta la larghezza dell'immagine.
 - ▶ `height`: Imposta l'altezza dell'immagine.
 - ▶ `scale`: Scala l'immagine.
 - ▶ `angle`: Ruota l'immagine di un certo angolo.
- ▶ **Posizionamento delle figure:**
 - ▶ Le figure possono essere posizionate con opzioni come `h` (here), `t` (top), `b` (bottom), `p` (page).

Inserimento di Immagini in L^AT_EX

Table

Le tabelle create con `\begin{tabular}...\end{tabular}` possono essere inserite in un ambiente “flottante” dato da

```
\begin{table}[t]
\centering
\begin{tabular}...\end{tabular}
\caption{Una tabella}
\end{table}
```

Introduzione ai Riferimenti Incrociati

I riferimenti incrociati in \LaTeX permettono di creare collegamenti tra diverse parti del documento. Utilizzando i comandi `\label` e `\ref`, puoi fare riferimento a:

- ▶ Sezioni
- ▶ Tabelle
- ▶ Figure
- ▶ Equazioni
- ▶ Teoremi

Riferimenti Incrociati per Sezioni

Per riferirti a una sezione, usa i comandi `\section` e `\label`.

Esempio di codice

```
\section{Introduzione}
```

```
\label{sec:introduzione}
```

```
Nella Sezione \ref{sec:introduzione} parleremo di...
```

Risultato: "Nella Sezione 1 parleremo di..."

Riferimenti Incrociati per Tabelle

Per riferirti a una tabella, utilizza i comandi `\begin{table}` e `\label`.

Esempio di codice

```
\begin{table}
\centering
\begin{tabular}{|c|c|}
...
\end{tabular}
\caption{Tabella di esempio}
\label{tab:esempio}
\end{table}
```

Come mostrato nella Tabella `\ref{tab:esempio}`...

Risultato: "Come mostrato nella Tabella 1..."

Riferimenti Incrociati per Figure

Per riferirti a una figura, utilizza i comandi `\begin{figure}` e `\label`.

Esempio di codice

```
\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{immagine.jpg}
\caption{Immagine di esempio}
\label{fig:esempio}
\end{figure}
```

Come illustrato nella Figura `\ref{fig:esempio}`...

Risultato: "Come illustrato nella Figura 1..."

Riferimenti Incrociati per Equazioni

Per riferirti a un'equazione, usa l'ambiente `equation` e il comando `\label`.

Esempio di codice

```
\begin{equation}
E = mc^2
\label{eq:energia}
\end{equation}
```

L'equazione `\eqref{eq:energia}` descrive la relazione tra energia e massa.

Risultato: "L'equazione (1) descrive la relazione tra energia e massa."

Riferimenti Incrociati per Teoremi

Per riferirti a un teorema, usa il pacchetto `amsthm` e il comando `\label`.

Esempio di codice

```
\usepackage{amsthm}
\newtheorem{teorema}{Teorema}

\begin{teorema}
\label{thm:esempio}
Ogni numero naturale è un intero.
\end{teorema}
```

Come affermato nel Teorema `\ref{thm:esempio}`...

Risultato: "Come affermato nel Teorema 1..."

Conclusion

I riferimenti incrociati in \LaTeX sono uno strumento potente per:

- ▶ Creare documenti ben organizzati.
- ▶ Facilitare la navigazione nel testo.
- ▶ Mantenere coerenza nelle citazioni.

Utilizzando `\label` e `\ref`, puoi migliorare la leggibilità e la facilità di manutenzione dei tuoi documenti \LaTeX , non devi preoccuparti di tenere traccia della numerazione!

Introduzione alla Bibliografia

In \LaTeX , la gestione della bibliografia può essere effettuata in due modi principali:

- ▶ **Ambiente** `thebibliography`: Un metodo semplice per gestire le citazioni.
- ▶ **Pacchetto** `biblatex`: Un metodo avanzato e flessibile per gestire le bibliografie.

Uso dell'Ambiente thebibliography

Per utilizzare l'ambiente thebibliography, seguire questi passaggi:

► Creazione della Bibliografia:

Esempio di codice

```
\begin{thebibliography}{99}
\bibitem{cite1}
Nome Autore, \textit{Titolo Libro}, Editore, Anno.

\bibitem{cite2}
Nome Autore, \textit{Titolo Articolo}, Rivista, Vol., pp.,
\end{thebibliography}
```

► Citazione nel Testo:

Esempio di codice

Come mostrato in `\cite{cite1}`...

Limitazioni dell'Ambiente thebibliography

L'ambiente thebibliography ha alcune limitazioni:

- ▶ Non supporta automaticamente stili di citazione diversi.
- ▶ Gestione manuale delle voci bibliografiche.
- ▶ Manca di funzionalità avanzate, come la gestione delle citazioni incrociate.

Uso del Pacchetto biblatex

Per una gestione avanzata della bibliografia, si consiglia di utilizzare biblatex. Ecco come utilizzarlo:

- ▶ **Importazione del pacchetto:**

Esempio di codice

```
\usepackage[backend=biber, style=apa]{biblatex}  
\addbibresource{bibliografia.bib}
```

Uso del Pacchetto biblatex

► Creazione del file bibliografico:

Esempio di codice (bibliografia.bib)

```
@book{cite1,  
author = {Nome Autore},  
title = {Titolo Libro},  
publisher = {Editore},  
year = {Anno},  
}  
  
@article{cite2,  
author = {Nome Autore},  
title = {Titolo Articolo},  
journal = {Rivista},  
volume = {Vol},  
pages = {pp},  
year = {Anno},  
}
```

Citazione nel Testo con biblatex

Dopo aver configurato biblatex, è possibile citare nel testo come segue:

Esempio di codice

Come mostrato in `\cite{cite1}`...

► **Stampa della Bibliografia:**

Esempio di codice

```
\printbibliography
```

Vantaggi di biblatex

Il pacchetto biblatex offre numerosi vantaggi:

- ▶ Supporta diversi stili di citazione.
- ▶ Gestisce automaticamente le voci bibliografiche.
- ▶ Permette citazioni incrociate e note a piè di pagina.
- ▶ Facilita l'aggiunta e la modifica di riferimenti.

In sintesi:

- ▶ `thebibliography` è semplice ma limitato.
- ▶ `biblatex` è potente e flessibile, adatto per documenti complessi.
- ▶ Scegliere il metodo giusto in base alle esigenze del documento.

I lucidi in \LaTeX , la classe Beamer

Beamer è un pacchetto per LaTeX che consente di creare presentazioni di alta qualità.

- ▶ Permette la creazione di slide professionali.
- ▶ Supporta vari temi e stili.
- ▶ Include funzionalità per l'inserimento di figure, tabelle, e collegamenti.

Creazione di Slide

Le slide in Beamer sono create utilizzando il comando `\begin{frame}` e `\end{frame}`.

Esempio di codice

```
\begin{frame}[opzioni]{Titolo delle slide}  
    Contenuto  
\end{frame}
```

Ogni `frame` rappresenta una slide nella presentazione.

Tipi di Slide

Beamer supporta diversi tipi di slide:

- ▶ **Frame Normali:** Contenuto generico.
- ▶ **Frame con Elenchi:** Utilizzo di elenchi puntati o numerati.
- ▶ **Frame con Figure e Tabelle:** Inserimento di immagini e tabelle.
- ▶ **Frame con Transizioni:** Animazioni per presentare il contenuto in sequenza.

Temi e Colorazioni

Beamer offre diversi temi e opzioni di colorazione:

- ▶ **Temi:** Cambiano l'aspetto generale della presentazione.
- ▶ **Colori:** Permettono di personalizzare i colori di sfondo, testo e accentuazione.

Esempio di codice per un tema

```
\usetheme{Madrid}
```

- ▶ È disponibile anche un **Tema Beamer del Dipartimento di Matematica:**

<https://www.dm.unipi.it/logo-del-dipartimento/>

Inserimento di Figure e Tabelle

Puoi facilmente inserire figure e tabelle nelle slide:

Esempio di codice per un'immagine

```
\begin{figure}[H]
\centering
\includegraphics[width=0.5\textwidth]{immagine.jpg}
\caption{Descrizione dell'immagine}
\label{fig:esempio}
\end{figure}
```

Inserimento di Figure e Tabelle

Puoi facilmente inserire figure e tabelle nelle slide:

Esempio di codice per una tabella

```
\begin{table}[H]
\centering
\begin{tabular}{|c|c|}
\hline
Colonna 1 & Colonna 2 \\
\hline
Dato 1 & Dato 2 \\
\hline
\end{tabular}
\caption{Tabella di esempio}
\label{tab:esempio}
\end{table}
```

Elementi Multimediali

Beamer consente anche di inserire elementi multimediali:

- ▶ **Video:** Supporta l'inserimento di file video.
- ▶ **Audio:** Puoi incorporare file audio nelle tue presentazioni.
- ▶ **Collegamenti ipertestuali:** Permette di aggiungere link a risorse esterne.

Animazioni

Beamer supporta animazioni e transizioni per rendere la presentazione più dinamica:

- ▶ **Overlay**: Permette di mostrare gradualmente il contenuto.

Esempio di codice per overlay

```
\begin{itemize}
\item<1-> Primo elemento
\item<2-> Secondo elemento
\end{itemize}
```

Animazioni

Beamer supporta animazioni e transizioni per rendere la presentazione più dinamica:

- ▶ **Overlay:** Permette di mostrare gradualmente il contenuto.
- ▶ **Transizioni:** Effetti tra le slide.

Esempio di codice per overlay

```
\begin{itemize}
\item<1-> Primo elemento
\item<2-> Secondo elemento
\end{itemize}
```

Esempio di utilizzo di `\only`

Ecco un esempio di utilizzo del comando `\only`:

Esempio di codice

```
\only<1->{Contenuto visibile nella slide 1 e nelle successive}
\only<2->{Contenuto visibile solo nella slide 2.}
```

In questo esempio:

- ▶ Il primo contenuto sarà visibile dalla slide 1 in poi.
- ▶ Il secondo contenuto sarà visibile solo nella slide 2.

Esempio di utilizzo di `onlyenv`

Ecco un esempio di utilizzo dell'ambiente `onlyenv`:

Esempio di codice

```
\begin{onlyenv}<1>
```

Questo testo è visibile solo nella slide 1.

```
\end{onlyenv}
```

```
\begin{onlyenv}<2>
```

Questo testo è visibile solo nella slide 2.

```
\end{onlyenv}
```

In questo esempio:

- ▶ Il primo blocco di testo sarà visibile solo nella slide 1.
- ▶ Il secondo blocco di testo sarà visibile solo nella slide 2.

Conclusioni

Abbiamo visto

- ▶ La storia di T_EX e L^AT_EX,
- ▶ Come produrre i documenti di base, inserire la matematica, tabelle e figure,
- ▶ Utilizzare gli automatismi per la gestione dei riferimenti incrociati e della bibliografia,
- ▶ Creare dei lucidi con contenuto matematico.

L^AT_EX può molto altro! Il documento di guida che trovate sul sito del corso e sull'E-Learning illustra altri dettagli e funzionalità.

Ci **eserciteremo in laboratorio** sui contenuti che abbiamo visto.