



EBook Gratuito

APPENDIMENTO

latex

Free unaffiliated eBook created from
Stack Overflow contributors.

#latex

Sommario

Di.....	1
Capitolo 1: Iniziare con il lattice	2
Osservazioni.....	2
LaTeX	2
Cos'è LaTeX?.....	2
Versioni.....	2
Examples.....	2
Installazione e configurazione.....	2
Installazione	3
Windows (TeXLive).....	3
Windows (MiKTeX).....	3
Mac OS X (TeXLive).....	3
Linux (TeXLive).....	3
Utilizzo dei gestori di pacchetti.....	3
Installazione da Upstream.....	4
Test di installazione	4
Editori LaTeX.....	5
Capitolo 2: Accesso alla documentazione dei pacchetti LaTeX	7
Examples.....	7
CTAN.....	7
TeX Live - texdoc.....	10
Capitolo 3: Aggiungi citazione	11
Examples.....	11
Aggiungi citazione al documento LaTeX già esistente.....	11
Capitolo 4: Classi di documenti	12
Sintassi.....	12
Osservazioni.....	12
Examples.....	12
Articolo.....	12

Quando utilizzare la classe di articoli?	12
Quali sono le specificità di questa classe?	12
Semplice esempio	12
Beamer	13
Quando usare la classe di beamer?	13
Quali sono le specificità di questa classe?	13
Semplice esempio	13
Definire la classe del documento	13
Capitolo 5: Composizione di matematica	14
introduzione	14
Sintassi	14
Osservazioni	14
Examples	14
Equazioni di base	14
Trovare simboli	15
Pacchetti disponibili per l'uso	16
Buoni comandi da sapere	17
Creare nuovi simboli	18
matrici	19
Capitolo 6: Contatori, se dichiarazioni e cicli con lattice	20
Examples	20
Contronotifica, inizializzazione e stampa in pdf	20
Operazioni con i contatori	20
Se affermazioni	22
Loop - Ripetendo le cose	23
Usare i loop in Tikz	24
Capitolo 7: Costruisci strumenti	26
Examples	26
Arara	26
Capitolo 8: Creare una bibliografia	28
Sintassi	28

Parametri.....	28
Examples.....	28
Bibliografia di base senza pacchetti (formattazione manuale).....	28
Bibliografia di base con biber.....	29
Capitolo 9: Creazione di poster con beamer.....	31
introduzione.....	31
Examples.....	31
Orientamento e dimensioni.....	31
Schema di base di un poster del proiettore.....	31
Esempio completo di poster di beamer.....	33
Capitolo 10: Definizione di macro.....	35
Sintassi.....	35
Parametri.....	35
Examples.....	35
Definizione di base di macro.....	35
Definire un nuovo comando di base.....	35
Definire un nuovo comando con argomenti.....	35
Ridefinire un comando esistente.....	36
Capitolo 11: Formattazione del testo.....	37
Examples.....	37
Emphazise Text.....	37
Colpisci il testo.....	37
Testo grassetto.....	37
Capitolo 12: Grafici di disegno.....	38
Examples.....	38
TikZ - Layout manuale.....	38
TikZ - Specifiche del grafico.....	38
TikZ - Disegno grafico algoritmico.....	40
Diagramma di transizione di stato di una catena di Markov.....	41
Capitolo 13: Intestazione e piè di pagina.....	43
Examples.....	43
Usando i pacchetti fancyhdr e titoli.....	43

Numero pagina come CurrPage / TotalPages nel piè di pagina	44
Capitolo 14: Pagine del titolo	46
Osservazioni.....	46
Examples.....	46
Titolo pagina report standard.....	46
Capitolo 15: Presentazione con pacchetto beamer	47
Parametri.....	47
Osservazioni.....	47
Examples.....	47
Semplice diapositiva del titolo dell'autore	47
Diapositiva del titolo di autore e affiliazione multipla.....	48
Capitolo 16: Spartiti per incisioni	50
Examples.....	50
LilyPond.....	50
Capitolo 17: tabelle	53
Examples.....	53
L'ambiente tabulare.....	53
Tavolo da colorare.....	54
Titoli di coda	58

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [latex](#)

It is an unofficial and free latex ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official latex.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con il lattice

Osservazioni

LaTeX

Cos'è LaTeX?

LaTeX (pronuncia *lay-tech* o *lah-tekh*) è un linguaggio di markup per la composizione di documenti simile a come HTML è uno per i siti web.

LaTeX offre vantaggi rispetto agli editor di tipo What-You-See-Is-What-You-Get (WYSIWYG) come Microsoft Word perché con LaTeX fornisci il contenuto e LaTeX si occupa del layout. Separazione del contenuto dalla composizione dei risultati in documenti coerenti e ben formattati. Inoltre, poiché la marcatura LaTeX è in formato di testo normale (diversamente dai formati di file più complessi prodotti dagli editor WYSIWYG, come `.docx`), i file LaTeX sono leggeri e possono essere facilmente mantenuti sotto il [controllo della versione](#).

I documenti LaTeX vengono generalmente compilati in file PDF, in modo che l'uniformità del layout venga mantenuta tra i diversi visualizzatori e per la stampa.

LaTeX è particolarmente popolare nella scrittura accademica grazie al suo ricco supporto per le equazioni di composizione, le figure e le tabelle di riferimenti incrociati e le citazioni e le bibliografie.

Versioni

Versione	Data di rilascio
LaTeX 2.09	1985/09/01
LaTeX 2e	1994/06/01

Examples

Installazione e configurazione

Puoi scegliere tra le principali distribuzioni di LaTeX:

- [TeX Live](#) (Windows, Linux e OS X), la distribuzione standard e multiplatforma.
- [MacTeX](#) (Mac) Una versione pacchettizzata di TeX Live realizzata per OS X con alcuni strumenti specifici per Mac
- [MiKTeX](#) (Windows) Una distribuzione separata interamente

Tutte le distribuzioni sono più o meno equivalenti in un mondo ideale. TeX Live ha il vantaggio di essere disponibile su tutte le piattaforme e quindi ha un supporto della community molto migliore. MiKTeX può sfruttare le funzionalità specifiche di Windows. Per motivi di licenza, MiKTeX distribuirà anche alcuni pacchetti che TeX Live non utilizzerà.

In tutti i casi, si consiglia l'installazione completa. In particolare, utilizzando la funzione di download-on-command di MiKTeX si bloccheranno / arresteranno molti editor.

Installazione

Windows (TeXLive)

1. Scarica il più recente TeXLive `install-tl-windows.exe` dal loro [sito web](#) .
2. Esegui `install-tl-windows.exe` e segui le istruzioni.

Windows (MiKTeX)

1. Scarica il programma di installazione MiKTeX più recente dal loro [sito web](#) .
2. Esegui il programma di installazione e segui le istruzioni.

Mac OS X (TeXLive)

1. Scarica il MacTeX più recente dal loro [sito web](#) .
2. Esegui `MacTeX.pkg` e segui le istruzioni.

Linux (TeXLive)

Gli utenti Linux hanno due opzioni:

1. Installa tramite il gestore di pacchetti della tua distribuzione (di solito diversi rilasci dietro)
2. Installa da upstream (pubblicato annualmente, aggiornato spesso)

Utilizzo dei gestori di pacchetti

- Arch Linux: `pacman -S texlive-most`
- Debian / Ubuntu / Mint: `apt-get install texlive-full`
- Fedora: `yum install texlive`

Si noti che l'uso di questo metodo significa che si dipenderà dal manutentore del pacchetto per la distribuzione degli aggiornamenti. Questi pacchetti saranno spesso diversi rilasci dietro la distribuzione più recente, spesso a significare che mancano aggiornamenti critici. È quasi sempre meglio installare da upstream. Si noti inoltre che il gestore pacchetti della distribuzione probabilmente non riconoscerà l'installazione diretta e potrebbe provare a installarlo quando si installano altri pacchetti di supporto correlati.

Installazione da Upstream

1. Scarica il più recente TeXLive `install-tl-unx.tar.gz` dal loro [sito web](#) .
2. Estrai i file dall'archivio con `tar -zxvf install-tl-unx.tar.gz` .
3. Passare alla cartella scaricata con `cd install-tl-unx` .
4. Esegui `./install-tl` e segui le istruzioni.

TeXLive dovrebbe ora essere installato in `/usr/local/texlive/YEAR/` , dove `YEAR` è l'anno a quattro cifre (es. `2016`). In questo modo, è possibile avere più versioni di TeXLive una accanto all'altra e passare da una all'altra modificando la variabile `PATH`.

Apri questa cartella e controlla la cartella `bin` . Dovrebbe contenere una sottocartella, che (a seconda della piattaforma) sarà simile a `i386-linux` o `x86_64-linux` .

5. Aggiungi la cartella binaria di TeX Live al tuo percorso con

```
EXPORT PATH=/usr/local/texlive/YEAR/bin/PLATFORM:$PATH
```

dove `YEAR` è l'anno a quattro cifre (ad esempio `2016`) e `PLATFORM` è la tua piattaforma (ad esempio `x86_64-linux`).

Test di installazione

L'installazione di LaTeX è ora completa. Per testarlo, crea un nuovo file con il tuo editor di testo preferito, `test.tex` e aggiungi il seguente contenuto:

```
\documentclass{article}
\begin{document}
  Hello World!
\end{document}
```

Ora, apri la console o il terminale, vai alla cartella in cui hai salvato `test.tex` ed esegui

```
pdflatex test
```

(Tieni presente che il tuo editor potrebbe avere le funzionalità per eseguire ciò per te.)

Questo crea diversi nuovi file, incluso `test.pdf` . Questo è il documento di output, e assomiglia a questo:

Hello World!

- L'editor [Vim](#) con il plugin [LaTeX-suite](#) .
- [Texmaker](#) - un IDE LaTeX specializzato.
- [TeXstudio](#) - un altro IDE LaTeX.
- [TeXworks](#) : un altro IDE LaTeX.

Mentre utenti esperti di Emacs o Vim potrebbero voler attenersi al loro editor (i cui plugin forniscono una serie di funzionalità non disponibili altrove), un IDE specializzato potrebbe essere più facile da installare / utilizzare per i principianti. Gli ultimi tre della lista hanno una funzione di anteprima in cui è possibile vedere i risultati della compilazione del documento.

Inoltre, ci sono strumenti LaTeX online che possono essere utili ai principianti o alle persone che devono collaborare, ad esempio [ShareLaTeX](#) e [Overleaf](#) .

Leggi [Iniziare con il lattice online](#): <https://riptutorial.com/it/latex/topic/2269/iniziare-con-il-lattice>

Capitolo 2: Accesso alla documentazione dei pacchetti LaTeX

Examples

CTAN

Il [completo TeX Archive Network](#) (CTAN) è in effetti questo, *il* repository completo dei pacchetti LaTeX. La maggior parte se non tutti i pacchetti di qualità (e altro) sono lì, e tutti quelli buoni includono la documentazione.

1. Inserisci il nome del pacchetto nella barra di ricerca.



2. Seleziona il pacchetto dall'elenco.

Search biblatex

The search found 16 of 42 hits in 10ms.

1 2 3 Next

Package biblatex

Sophisticated Bibliographies in L^AT_EX

[/pkg/biblatex](#)

Last modified: 2016-05-15 17:33

Topic biblatex

BibL^AT_EX bibliography support

[/topic/biblatex](#)

Last modified: 2016-07-04 18:34

Package biblatex-jura

BibL^AT_EX stylefiles for German legal literature

[/pkg/biblatex-jura](#)

Last modified: 2015-08-03 06:36

Package biblatex-bwl

BibL^AT_EX citations for FU Berlin

[/pkg/biblatex-bwl](#)

Last modified: 2015-08-03 06:36

Package biblatex-nejm

BibL^AT_EX style for the New England Journal of Medicine (NEJM)

[/pkg/biblatex-nejm](#)

Last modified: 2015-08-03 06:36

3. Accedi ai documenti della documentazione.

BibL^AT_EX – Sophisticated Bibliographies in L^AT_EX

BibL^AT_EX is a complete reimplementaion of the bibliographic facilities provided by L^AT_EX. Formatting of the bibliography is entirely controlled by L^AT_EX macros, and a working knowledge of L^AT_EX should be sufficient to design new bibliography and citation styles. BibL^AT_EX uses its own data backend program called “[biber](#)” to read and process the bibliographic data. With [biber](#), BibL^AT_EX has many features rivalling or surpassing other bibliography systems. To mention a few:

- Full Unicode support
- Highly customisable sorting using the Unicode Collation Algorithm + CLDR tailoring
- Highly customisable bibliography labels
- Complex macro-based on-the-fly data modification without changing your data sources
- A tool mode for transforming bibliographic data sources
- Multiple bibliographies and lists of bibliographic information in the same document with different sorting
- Highly customisable data source inheritance rules
- Polyglossia and babel support for automatic language switching for bibliographic entries and citations
- Automatic bibliography data recoding (UTF-8 -> latin1, L^AT_EX macros -> UTF-8 etc)
- Remote data sources
- Highly sophisticated automatic name and name list disambiguation system
- Highly customisable data model so users can define their own bibliographic data types
- Validation of bibliographic data against a data model
- Subdivided and/or filtered bibliographies, bibliographies per chapter, section etc.

Apart from the features unique to BibL^AT_EX, the package also incorporates core features of the following packages: [babelbib](#), [bibtopic](#), [bibunits](#), [chapterbib](#), [cite](#), [inlinebib](#), [mcite](#) and [mciteplus](#), [mlbib](#), [multibib](#), [splitbib](#).

Sources	/macros/latex/contrib/biblatex
Documentation	 Readme   Package documentation (English)   Release notes for current version 
Version	3.4
License	The L^AT_EX Project Public License 1.3
Copyright	2012–2016 Philipp Lehman, Joseph Wright, Audrey Boruvka, Philip Kime 2006–2012 Philipp Lehman
Maintainer	Philipp Lehman (inactive) Philip Kime
Contained in	T_EX Live as biblatex MiK_TE_X as biblatex
Topics	BibL^AT_EX bibliography support bibliography processor



[Download](#) the contents of this package in one zip archive (17.2M).

Importante: CTAN contiene le versioni più recenti. Se la tua installazione non è aggiornata, la documentazione non corrisponderà! In tal caso, consultare i documenti di documentazione forniti

con la propria distribuzione LaTeX.

TeX Live - texdoc

Se si utilizza la distribuzione di [TeX Live](#) è possibile utilizzare il programma da riga di comando `texdoc` . Per esempio,

```
texdoc biblatex
```

si aprirà la documentazione del pacchetto `biblatex` .

O se non sei esperto di riga di comando, lo stesso si può trovare online su <http://www.texdoc.net/>

Leggi [Accesso alla documentazione dei pacchetti LaTeX online:](#)

<https://riptutorial.com/it/latex/topic/5820/accesso-alla-documentazione-dei-pacchetti-latex>

Capitolo 3: Aggiungi citazione

Examples

Aggiungi citazione al documento LaTeX già esistente

Alla fine del documento aggiungi quanto segue:

`\bibliographystyle { style }`

`\bibliografia { posizione del file }`

Creare un file con estensione *.bib* e salvare la citazione come segue:

```
@inproceedings{citation_name,  
  title={Paper Title},  
  author={List Authors},  
  pages={45--48},  
  year={2013},  
  organization={organization name}  
}
```

Per citare utilizzare il seguente: **`\CITET {} citation_name`**

Leggi Aggiungi citazione online: <https://riptutorial.com/it/latex/topic/8357/aggiungi-citazione>

Capitolo 4: Classi di documenti

Sintassi

- \Documentclass {...}

Osservazioni

Questo argomento mira a spiegare i diversi tipi di documento e le loro specificità.

Un buon modo per organizzarlo sarebbe 1 esempio per tipo

Examples

Articolo

```
\documentclass{article}
```

Quando utilizzare la classe di articoli?

Per articoli su riviste scientifiche, presentazioni, brevi rapporti, documentazione del programma, inviti, ... ¹

Quali sono le specificità di questa classe?

Un articolo non contiene capitoli o parti. Può essere diviso in sezioni, sottosezioni e paragrafi ecc.

Per impostazione predefinita, il titolo viene visualizzato nella parte superiore della prima pagina e non in una pagina del titolo separata.

Semplice esempio

```
\documentclass{article}

\title{Hello world}
\author{Me    }
\date{\today}

\begin{document}

\maketitle

Hello, World!
```

```
\end{document}
```

Beamer

```
\documentclass{beamer}
```

Quando usare la classe di beamer?

Per le diapositive della presentazione.

Quali sono le specificità di questa classe?

L'output è orientato al paesaggio. Il documento è separato in "frame" (diapositive).

Semplice esempio

L'esempio seguente è stato adattato da: sharelatex.com/learn/Beamer

```
\documentclass{beamer}

\usepackage[utf8]{inputenc}

\title{Sample title}
\author{Me}
\date{\today}

\begin{document}

\frame{\titlepage}

\begin{frame}
\frametitle{Sample frame title}
This is a text in first frame. This is a text in first frame. This is a text in first frame.
\end{frame}

\end{document}
```

Definire la classe del documento

La prima riga in ciascuno dei tuoi programmi LaTeX dovrebbe farlo. Dovrebbe seguire la forma `\documentclass{...}`. Quello che metti tra le parentesi graffe è molto importante. Alcune classi di documenti ti forniscono comandi aggiuntivi da usare, altri usano un formato diverso e tutti hanno parametri specifici che puoi inserire (descritti nella sezione parametri).

Leggi **Classi di documenti online**: <https://riptutorial.com/it/latex/topic/7458/classi-di-documenti>

Capitolo 5: Composizione di matematica

introduzione

Uno dei maggiori vantaggi di LaTeX è la sua abilità nella composizione di equazioni. Qui sono descritti i fondamenti delle equazioni di composizione, alcuni dei vari pacchetti che possono essere utilizzati, nonché i simboli comuni.

Sintassi

- `\begin {equation} ... \end {equation}`
- `testo $... $ testo`
- `\usepackage {amsmath} ... \begin {equation *} ... \end {equation *}`

Osservazioni

Ecco alcune idee di base per assicurarti che il tuo codice non si interrompa e le tue equazioni siano migliori:

1. Assicurati che tutte le parentesi, le parentesi graffe, i simboli del dollaro e i comandi `\begin{}` `\end{}` corrispondano. Questo è qualcosa in cui un piccolo errore può rovinare tutto il tuo codice in grande stile.
2. Se ricevi errori, assicurati di aver caricato il pacchetto corretto (ad esempio, non utilizzare il comando `\begin{equation*}` senza il pacchetto `amsmath`).
3. Mai, mai, *mai* usare i doppi simboli del dollaro (`$$an equation here$$`) invece di `\begin{equation}` .
4. Non usare mai la modalità matematica come modo per rendere il testo in corsivo.
5. Completamente bloccato? Prova [TeX.SX](https://tug.ctan.org/info/TeX.SX/) , un sito per rispondere a domande su TeX, LaTeX e le lingue correlate.

In bocca al lupo!

Examples

Equazioni di base

Equazioni semplici e in linea

Puoi fare una semplice equazione in linea usando `$an equation here$` .

Ad esempio, potresti farlo

```
 $\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar{z}$
```

che, se mettiamo un piccolo testo falso intorno, dà

Foo $\lim_{n \rightarrow \infty} \frac{1}{2^n} i \bar{z}$ quux

Equazioni numerate e centrate

Quando si scrivono documenti o altri documenti, a volte è preferibile avere le equazioni centrate e numerate, anziché in linea. Quindi, usa i comandi `\begin{equation}` e `\end{equation}`.

Ad esempio, se usiamo il codice

```
\begin{equation}
\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar{z}
\end{equation}
```

E aggiungere un piccolo testo intorno, otteniamo

Foo quux bla ipsum ipsum $\lim_{n \rightarrow \infty} \frac{1}{2^n} i \bar{z}$ foo quux bla.

Puoi rimuovere la numerazione dell'equazione usando `\begin{equation*}` e `\end{equation*}`.

Ad esempio, se usiamo il codice

```
\begin{equation*}
\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar{z}
\end{equation*}
```

e aggiungere un piccolo testo intorno ad esso, otteniamo

Foo quux bla bla ipsum $\lim_{n \rightarrow \infty} \frac{1}{2^n} i \bar{z}$
quux.

(sebbene si noti che devi usare il pacchetto `amsmath` per questo).

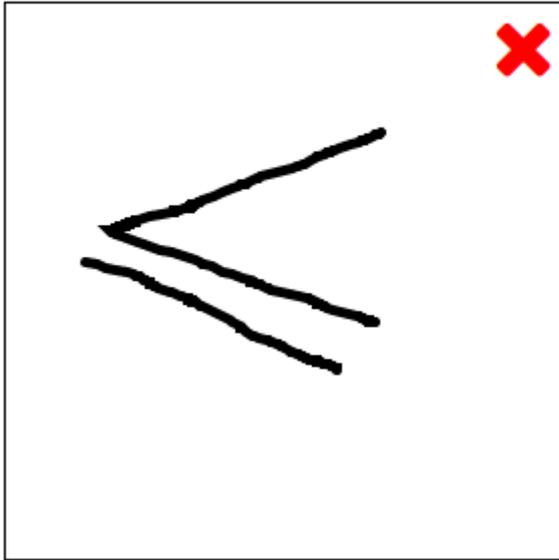
Trovare simboli

A volte, può essere difficile trovare il simbolo matematico di cui hai bisogno. Ci sono diverse opzioni qui. Il primo (e il più veloce) è usare [Detexify](#), dove si disegna il simbolo che si desidera, e cerca di trovare quello che vuoi, come mostrato di seguito:

Detexify

classify

symbols



Score: 0.076721263121454
`\usepackage{ amssymb }`
`\leqslant`
mathmode



Score: 0.11561454413754882
`\usepackage{ amssymb }`
`\leq`
mathmode



Score: 0.11892004279045627
`\usepackage{ tipa }`
`\text{rptr}`
textmode

Un'altra opzione è quella di utilizzare l'esauriente elenco di simboli LaTeX, che può essere trovato [qui](#) . Se si utilizza il pacchetto `unicode-math` [questo elenco](#) di tutti i simboli supportati può essere utile. Un'altra opzione è [questo sito web](#) , che ha simboli matematici comuni.

Pacchetti disponibili per l'uso

Mentre LaTeX standard è tutto ciò che è necessario per la maggior parte delle semplici formule matematiche ed equazioni, a volte sono necessari più simboli e strumenti. Sono disponibili più pacchetti che miglioreranno le equazioni e ti forniranno più cose con cui lavorare. Tre dei pacchetti principali sono descritti di seguito. Ricorda, per caricare un pacchetto, digita `\usepackage{package}` nel preambolo del documento.

`amsmath`

Il pacchetto `amsmath` è un pacchetto incredibilmente utile. È usato per permettere che le equazioni siano centrate ma non numerate, come in `\begin{equation*}` , è usato per creare matrici (come descritto sotto) e introduce molti altri comandi utili, come `\overset` e `\underset` , descritto sotto. La documentazione del pacchetto `amsmath` può essere trovata [qui](#) .

`mathtools`

Il pacchetto `mathtools` basa sul pacchetto `amsmath` , aggiungendo ulteriori simboli e strumenti utili. Carica automaticamente il pacchetto `amsmath` , quindi non è necessario caricarli entrambi nel preambolo del documento. La documentazione di `mathtools` può essere trovata [qui](#) .

`amssymb`

Il pacchetto `amssymb` offre molti simboli extra che possono essere molto utili per equazioni più complesse. La documentazione di `amssymb` può essere trovata [qui](#) .

Pacchetti di font

Ci sono anche vari tipi di caratteri che puoi usare per le tue equazioni, come descritto su [questa domanda](#) sullo scambio di stack TeX, per TeX, LaTeX e gli amici.

[Questo documento](#) è una spiegazione concisa delle diverse funzionalità fornite da alcuni pacchetti e da LaTeX standard; è molto utile.

Buoni comandi da sapere

Alcuni dei comandi più comuni includono:

- **Frazioni e radici quadrate:** per le frazioni, usa `\frac {numerator}{denominator}` . Per radici quadrate, utilizzare `\sqrt [root]{number}` .
- **Lettere greche:** usa i comandi indicati nella tabella seguente:

αA	<code>\alpha A</code>	νN	<code>\nu N</code>
βB	<code>\beta B</code>	$\xi \Xi$	<code>\xi \Xi</code>
$\gamma \Gamma$	<code>\gamma \Gamma</code>	$o O$	<code>o O</code>
$\delta \Delta$	<code>\delta \Delta</code>	$\pi \Pi$	<code>\pi \Pi</code>
$\epsilon \varepsilon E$	<code>\epsilon \varepsilon E</code>	$\rho \varrho P$	<code>\rho \varrho P</code>
ζZ	<code>\zeta Z</code>	$\sigma \Sigma$	<code>\sigma \Sigma</code>
ηH	<code>\eta H</code>	τT	<code>\tau T</code>
$\theta \vartheta \Theta$	<code>\theta \vartheta \Theta</code>	$\upsilon \Upsilon$	<code>\upsilon \Upsilon</code>
ιI	<code>\iota I</code>	$\phi \varphi \Phi$	<code>\phi \varphi \Phi</code>
κK	<code>\kappa K</code>	χX	<code>\chi X</code>
$\lambda \Lambda$	<code>\lambda \Lambda</code>	$\psi \Psi$	<code>\psi \Psi</code>
μM	<code>\mu M</code>	$\omega \Omega$	<code>\omega \Omega</code>

- **Operatori:** `\leq` restituisce il simbolo minore o uguale a, `\geq` dà il maggiore o uguale al simbolo, `\neq` dà il simbolo non uguale, `\sum` dà il simbolo sommatoria, `\partial` dà il simbolo derivativo parziale, `\nabla` dà l'operatore di Laplacian, `\times` fornisce il simbolo del prodotto o della moltiplicazione incrociati, `\cdot` fornisce il simbolo del prodotto o della moltiplicazione, e `\int` fornisce il simbolo integrale.
- **Frecce:** `\rightarrow` e `\leftarrow` danno rispettivamente le frecce destra e sinistra.
- **Percentuali:** se si digita% in LaTeX, è importante includere una barra rovesciata, `\%` poiché il simbolo di percentuale viene normalmente utilizzato per i commenti.
- **Superscripts e subscript:** per eseguire un apice, è possibile digitare `x^2` o, per apici più

lunghi, x^{2x} . Per fare un pedice, puoi digitare x_a , o, per gli abbonati più lunghi, x_{ab} .

- **Grassetto:** usa `\boldmath{...}` per rendere i tuoi simboli matematici in grassetto. Altre opzioni sono fornite in [questa domanda di TeX.SX](#). I simboli matematici sono automaticamente in corsivo; se non vuoi che questo sia vero, crea il testo della tua equazione come descritto di seguito.
- **Infinito:** per scrivere infinito, usa il comando `\infty`.
- **Spostare gli oggetti sopra o sotto un altro:** in primo luogo, solo per gli operatori matematici, esiste un metodo alternativo. È possibile digitare l'operatore matematico, pronunciare `\int` e quindi utilizzare il comando `\limits`. Un esempio è `\int\limits_{\infty}` o `\int\limits^{\infty}`. Quindi, per i casi normali, puoi eseguire `\overset{top}{normal}` o `\underset{bottom}{normal}`. Questo può essere molto utile per fare i vettori. Ad esempio, potresti eseguire `\vec{x}` Il pacchetto `amsmath` è necessario per `overset` e `underset`.
- **Bretelle ricce:** poiché le parentesi graffe vengono utilizzate nei comandi, è necessario digitare `\{ o \}` per ottenere parentesi graffe.
- **Testo:** per includere il testo nelle equazioni, digitare `\usepackage{amsmath}` nel preambolo, quindi digitare `\text{...}`.
- **Spazio:** per aggiungere spazio nelle equazioni, digitare `\quad` tra i due elementi che si desidera separare (ad esempio, potresti avere $2x \quad \cos$).

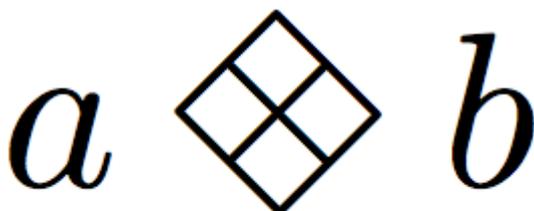
Creare nuovi simboli

Diciamo che non riesci a trovare il simbolo di cui hai bisogno da nessuna parte. È possibile creare un simbolo personalizzato. Ad esempio, il codice

```
\documentclass{article}
\usepackage{graphicx,amsmath,amssymb}
\DeclareRobustCommand{\diamondtimes}{%
  \mathbin{\text{\rotatebox[origin=c]{45}{\boxplus}}}%
}

\begin{document}
$a\diamondtimes b$
\end{document}
```

crea e chiama un simbolo, dando



Questo è un esempio più semplice; deve semplicemente ruotare un simbolo già esistente. Tuttavia, puoi creare simboli più complessi.

Questa sezione è in fase di espansione.

matrici

matrici

Devi sempre usare il pacchetto `amsmath` se stai per usare i seguenti comandi. Esistono quattro tipi principali di matrice, come mostrato nel seguente codice:

```
\begin{matrix}
  a & b \\
  c & d
\end{matrix}
\quad
\begin{pmatrix}
  a & b \\
  c & d
\end{pmatrix}
\quad
\begin{bmatrix}
  a & b \\
  c & d
\end{bmatrix}
\quad
\begin{vmatrix}
  a & b \\
  c & d
\end{vmatrix}
\quad
\begin{Vmatrix}
  a & b \\
  c & d
\end{Vmatrix}
```

Questo codice produce

$$\begin{matrix} a & b \\ c & d \end{matrix} \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

Ci sono un paio di cose importanti da notare su questo:

1. È importante inserire la matrice all'interno `equation`, `equation*` o `$...$` ambiente - il comando `bmatrix` non è un ambiente matematico a sé stante.
2. La costruzione della matrice è in realtà abbastanza semplice. Per ogni riga, crei ciascun elemento (ad esempio `x_{11}`), quindi inserisci un `&` e quindi scrivi l'elemento successivo. Per più righe, alla fine di ogni riga inserisci `\\` (non devi farlo per l'ultima riga). È abbastanza simile a un tavolo in questo.

Leggi [Composizione di matematica online](https://riptutorial.com/it/latex/topic/5950/composizione-di-matematica): <https://riptutorial.com/it/latex/topic/5950/composizione-di-matematica>

Capitolo 6: Contatori, se dichiarazioni e cicli con lattice

Examples

Contronotifica, inizializzazione e stampa in pdf

È possibile utilizzare variabili intere con latex. Per creare una nuova variabile abbiamo bisogno del `\newcounter{name}`, dove `name` è il nome del nuovo contatore. Il `name` deve contenere solo lettere. Questo comando ne crea uno nuovo con nome `\thename`. Con questo comando possiamo stampare la variabile del `name` sulla carta. Il valore iniziale del `name` è 0. Per dare valore a "nome" possiamo usare `\setcounter{name}{n}` dove `n` è un numero intero. `\value{name}` è una funzione che restituisce il valore del `name`.

```
\documentclass{article}
\begin{document}
\newcounter{num}           %new counter, initial value is 0
\thenum                   %print 0
\setcounter{num}{3}       %set num to 3
\thenum                   %print 3
\newcounter{number}
\setcounter{number}{\value{num}} %set number to value of num
\thenumber                %print 3
```

Latex provides some other formats to print a number.

Other types of printing:

```
\arabic{num} \\
\Roman{num} \\ %→ I, II, III, IV, . . . (num = 1, 2, 3, . . . )
\roman{num} \\ %→ i, ii, iii, iv, . . . (num = 1, 2, 3, . . . )
\Alph{num} \\ %→ A, B, C, D, . . . (num = 1, 2, 3, . . . , 26)
\alph{num} \\ %→ a, b, c, d, . . . (num = 1, 2, 3, . . . , 26)
\fnsymbol{num} \\ %→ *, †, ‡, §, ¶, k, **, ††, ‡‡ (num = 1, 2, 3, . . . , 9)
\end{document}
```

0

3

3

Latex provides some other formats to print a number.

Other types of printing:

3

III

iii

C

c

‡

Operazioni con i contatori

Questo esempio mostra come utilizzare le operazioni matematiche con i contatori. Può essere utile per i loop in lattice.

Aggiunta: `\addtocounter{num}{n}`

questo comando aggiunge n a num , dove num è un contatore e n è un numero intero positivo.

Sottrazione: `\addtocounter{num}{-n}`

questo comando sottrae n da num , dove num è un contatore e n è un numero intero positivo.

Moltiplicazione: `\multiply\value{num} by n`

questo comando moltiplica num per n , dove num è un contatore e n è un numero intero.

Divisione `\divide\value{num} by n`

questo comando divide num per n e ottiene la parte intera del quoziente (num è un contatore e n è un numero intero)

```
\documentclass{article}
  \begin{document}
  \newcounter{num}
  \setcounter{num}{3}
  \addtocounter{num}{10}
  \thenum\%prints 13
  \addtocounter{num}{-3}
  \thenum\%prints 10
  \stepcounter{num}
  \thenum\%prints 11
  \multiply\value{num} by \value{num}
  \thenum\%prints 121
  \multiply\value{num} by 2
  \thenum\%prints 242
  \divide\value{num} by 60
  \thenum%prints 4
\end{document}
```

`\newcommand{num}` dichiara contatore. `\setcounter{num}{3}` imposta il valore numerico su 3.

`\addtocounter{num}{10}` aggiunge 10 a num .

`\addtocounter{num}{-3}` sottrarre 3 da num .

`\stepcounter{num}` aggiunge 1 a num

`\multiply\value{num} by \value{num}` quadrati num .

`\multiply\value{num} by 2` numeri doppio.

`\divide\value{num} by 60` divide num per 60 e ottiene la parte intera.

Il risultato del codice: 13 \ 10 \ 11 \ 121 \ 242 \ 4

(\\ simbolizza la nuova riga)

Il pacchetto intcalc aggiunge alcune altre operazioni su interi come mod, pow, sng, abs, inv ...

[intcalc_package.pdf](#)

Se affermazioni

Nel latex possiamo usare i comandi integrati per eseguire il codice sia che le condizioni siano vere o no.

Confronto tra due numeri interi: `\ifnum\value{num}>n {A} \else {B}\fi`

Questo codice esegue A se $num > n$ else B. Possiamo sostituire $>$ con $<e =$.

Se un numero è dispari: `\ifodd\value{num} {A}\else {B}\fi`

Se num è dispari, esegue A else else.

Se con condizione: `\ifthenelse{condition}{A}{B}`

Dobbiamo caricare il pacchetto ifth per utilizzare questo comando. Se la condizione è vera, esegue A else B.

È possibile creare condizioni complesse con `\(\)`, `\AND`, `\OR`, `\NOT`.

Ad esempio: `\ifthenelse{\(\NOT 4<2 \OR 4>11\)\AND\isodd{4}}{A}{B}`

Questo pezzo di codice scrive "B" sulla pagina. `\NOT 4<2` è vero e `4>11` è falso. Se colleghiamo un'istruzione falsa e una vera con "OR", il risultato è vero. Quindi `\(\NOT 4<2 \OR 4>11\)` è vero. `\isodd{4}` è falso perché 4 è pari. Una falsa e una vera affermazione connessa a "AND" è falsa, quindi l'output è B.

Un codice di esempio:

```
\documentclass{article}
\usepackage{ifthen}
\begin{document}
  \newcounter{num}
  \setcounter{num}{10}

  If num$>$100 then the next sentence will be "Num is large." else "Num is small."

  Num is \ifnum \value{num}>100 {large} \else {small}.

  If num is odd then the next sentence will begin with "Odd" if not then with "Even"

  \ifodd \value{num} {Odd} \else {Even} numbers are cool.

  If (num$>$3 and (1$<$0 or num$=$10)) is true then the next sentence will be "True." else
  "False."

  \ifthenelse{\value{num}>3\AND\ (1<0 \OR \value{num}=10\)}{True.}{False.}
```

```
\end{document}
```

```
If num>100 then the next sentence will be "Num is large." else "Num is
small."
  Num is small.
  If num is odd then the next sentence will begin with "Odd", if not then with
  "Even"
  Even numbers are cool.
  If (num>3 and (1<0 or num=10)) is true then the next sentence will be
  "True." else "False."
  True.
```

Loop - Ripetendo le cose

Possiamo creare loop in lattice. Sono simili ma non così personalizzabili come loop in altri linguaggi di programmazione. Un'alternativa ai loop di utilizzo è `@loops`. Se usiamo un comando che include "@" nel suo nome, dobbiamo metterlo tra `\makeatletter` e `\makeatother`. Non è consentito utilizzarli in una macro che descrive una nuova definizione.

Sbagliato:

```
\def\is#1#2{\makeatletter\ifstar{#1}{#2}\makeatother
```

Destra:

```
\makeatletter\def\is#1#2{\@ifstar{#1}{#2}}\makeatother
```

@for loop: `\@for\command:={list}\do{commands}`

Esempio :

```
\makeatletter
\@for\sun:={rising,setting}\do{The sun is \sun.}
\makeatother
```

Crea il seguente testo: Il sole sta sorgendo. Il sole sta tramontando.

@whilenum loop: `\@whilenum condition\do{commands}`

Esempio :

```
\makeatletter
\newcounter{int}
\@whilenum\value{int}<10\do
{\stepcounter{int}\ifthenelse{\isodd{\value{int}}}{\theint}{}}
\makeatother
```

Questo codice scrive numeri dispari da 1 a 9.

ciclo "loop repeat": `\loop {commands} \ifnum condition \repeat`

Esegue i comandi finché la condizione è vera.

Esempio

```
\setcounter{int}{1}
\loop
\theint
\addtocounter{int}{2}
\ifnum \value{int}<10
\repeat
```

Questo codice fa lo stesso del ciclo @whilenum.

Un codice di esempio:

```
\documentclass{article}
\usepackage{ifthen}
\usepackage{amsmath} %\text{} command needs this package
\begin{document}
  Demonstration of @for loop:

  \makeatletter
  \@for\sun:={rising,setting}\do{The sun is \sun. }
  \makeatother

  \newcounter{int}

  @whilenum loop:

  \setcounter{int}{0}
  \makeatletter
  \@whilenum\value{int}<20\do
  {\stepcounter{int}\ifthenelse{\isodd{\value{int}}}{\theint\text{ }}{}}
  \makeatother

  "loop repeat" loop:

  \setcounter{int}{1}
  \loop
  \theint
  \text{ }\addtocounter{int}{2}\ifnum\value{int}<20
  \repeat
\end{document}
```

```
Demonstration of @for loop:
The sun is rising. The sun is setting.
@whilenum loop:
1 3 5 7 9 11 13 15 17 19
"loop repeat" loop:
1 3 5 7 9 11 13 15 17 19
```

Usare i loop in Tikz

I loop sono utili in Tikz.

Il seguente codice disegna un orologio senza numeri:

```
\documentclass{article}
\usepackage{ifthen}
```

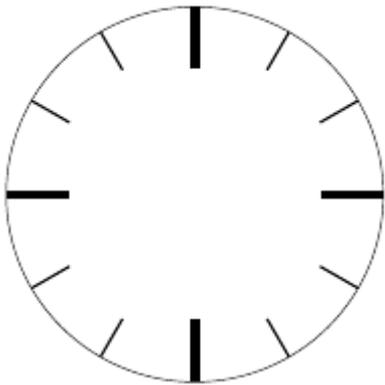
```

\usepackage{intcalc}
\usepackage{tikz}
\newcounter{num}

\begin{document}
\begin{tikzpicture}
  \makeatletter
  \setcounter{num}{1}
  \newcounter{angle}
  \draw (0,0) circle (3cm);
  \@whilenum\value{num}<13\do{
    \setcounter{angle}{360}
    \multiply\value{angle} by \value{num}
    \divide\value{angle} by 12
    \ifnum \intcalcMod{\value{num}}{3}=0{
      \draw[line width=4pt] (\theangle:2cm) -- (\theangle:3cm);    } \else
    {
      \draw[line width=1pt] (\theangle:2.3cm) -- (\theangle:3cm);
    } \fi
    \addtocounter{num}{1}
  }
  \makeatother
\end{tikzpicture}
\end{document}

```

Il risultato:



Leggi Contatori, se dichiarazioni e cicli con lattice online:

<https://riptutorial.com/it/latex/topic/9224/contatori--se-dichiarazioni-e-cicli-con-lattice>

Capitolo 7: Costruisci strumenti

Examples

Arara

Arara è uno strumento di automazione multiplatforma appositamente progettato per TeX. È incluso in una distribuzione standard, quindi non è necessario installare nulla di aggiuntivo. È più efficacemente inteso come mezzo per registrare le istruzioni di compilazione nel file TeX stesso:

```
% arara: pdflatex
\documentclass{article}
\begin{document}
  Hello, world
\end{document}
```

Questi possono essere molto più complicati, però:

```
% arara: pdflatex
% arara: biber
% arara: pdflatex

% To support a self-contained example, this builds a BibTeX file on-the-fly
\begin{filecontents}{references.bib}
@article{dijkstra,
  author = {Dijkstra, Edsger W.},
  title = {Self-stabilizing Systems in Spite of Distributed Control},
  journal = {Commun. ACM},
  issue_date = {Nov. 1974},
  volume = {17},
  number = {11},
  month = nov,
  year = {1974},
  issn = {0001-0782},
  pages = {643--644},
  numpages = {2},
  url = {http://doi.acm.org/10.1145/361179.361202},
  doi = {10.1145/361179.361202},
  acmid = {361202},
  publisher = {ACM},
  address = {New York, NY, USA},
  keywords = {distributed control, error recovery, harmonious cooperation, multiprocessing,
mutual exclusion, networks, robustness, self-repair, self-stabilization, sharing,
synchronization},
}
\end{filecontents}

\documentclass{article}
\usepackage[backend=biber]{biblatex}
\addbibresource{references.bib}

\begin{document}
Hello, World! \cite{dijkstra}.
```

```
\printbibliography  
\end{document}
```

Leggi **Costruisci strumenti online**: <https://riptutorial.com/it/latex/topic/5015/costruisci-strumenti>

Capitolo 8: Creare una bibliografia

Sintassi

- Per una bibliografia formattata manualmente, non è necessario avere citazioni - `\cite` - all'interno del documento.

Parametri

Parametro	Dettaglio
<code>thebibliography</code>	Questo ambiente definisce lo scopo della bibliografia attuale. Definisce un ambiente di tipo lista all'interno del quale puoi usare <code>\bibitem</code> per impostare un elemento bibliografico.
<code>{x}</code>	L'ambiente <code>thebibliography</code> accetta un singolo argomento che rappresenta l'elemento più ampio da aspettarsi nell'enumerazione di <code>\bibitem</code> s. Per meno di 10 voci, utilizzare un singolo carattere / cifra; per meno di 100 voci, utilizzare due caratteri / cifre, ...
<code>\bibitem{<a>}</code>	Imposta l'elemento della bibliografia <code></code> e rendilo disponibile per <code>\cite</code> all'interno del documento usando l'etichetta <code><a></code> .

Examples

Bibliografia di base senza pacchetti (formattazione manuale)

See [1] or [2] or [1, 2].

References

[1] AUTHOR, A, *A title*, Journal of So-and-So, 2000.

[2] SOMEONE, B, *Another title*, Book of books, 1900.

```
\documentclass{article}% or book, report, ...

\begin{document}

See \cite{citeA} or \cite{citeB} or \cite{citeA, citeB}.

\begin{thebibliography}{x}
% \bibitem{<biblabel>} <citation>

```

```

\bibitem{citeA}
  {\scshape Author, A}, {\itshape A title}, Journal of So-and-So, 2000.
\bibitem{citeB}
  {\scshape Someone, B}, {\itshape Another title}, Book of books, 1900.
\end{thebibliography}

\end{document}

```

Nota che a meno che tu non sappia davvero *perché*, probabilmente non dovresti farlo. È preferibile l'uso di pacchetti designati (vedi altri esempi).

Bibliografia di base con biber

Per iniziare una bibliografia devi definire le tue fonti. Creare un [file di database](#) (come `sources.bib`) e includere alcuni contenuti:

```

@book{Doe1993,
  Author = {John Doe},
  Publisher = {Earth University},
  Title = {Creating a bibliography with biber},
  Year = {1993}}

```

È quindi possibile includere il file di database nel documento principale e citare la nuova fonte (`Doe1993`).

```

\documentclass{article}

% Include the biblatex package and tell it to use biber as a backend.
% Without specifying the backend, it assumes biber.
\usepackage[backend=biber]{biblatex}

% Define where biber can find your sources
\addbibresource{sources.bib}

\begin{document}
"Biber isn't that difficult." \cite{Doe1993}
% Use \cite{source-ID} to generate a citation

% Print the bibliography
\printbibliography

\end{document}

```

Per compilare il documento, dovrai eseguire 3 comandi in sequenza:

1. `pdflatex` per creare un file ausiliario che dice a biber quali sono le fonti necessarie
2. `biber` per creare un file ausiliario con tutte le fonti che possono essere utilizzate da `pdflatex`
3. `pdflatex` per includere il file ausiliario e creare il PDF

"Biber isn't that difficult." [1]

References

- [1] John Doe. *Creating a bibliography with biber*. Earth University, 1993.

Trova molte altre opzioni e campi aggiuntivi per i file bib nella [documentazione](#) del [pacchetto su CTAN](#) .

Leggi [Creare una bibliografia online](#): <https://riptutorial.com/it/latex/topic/3488/creare-una-bibliografia>

Capitolo 9: Creazione di poster con beamer

introduzione

Creare un poster usando il pacchetto beamerposter è molto simile alla creazione di un singolo frame. Metti il contenuto in colonne. All'interno di ciascuna colonna, separa il contenuto utilizzando i blocchi.

Examples

Orientamento e dimensioni

Durante l'aggiunta del pacchetto beamerposter, fornire i parametri richiesti.

```
\usepackage[orientation=landscape,size=a1]{beamerposter}
```

Puoi anche personalizzare la dimensione del poster.

```
\usepackage[orientation=portrait,size=custom,height=110,width=80,scale=1.4]{beamerposter}
```

Le dimensioni di altezza e larghezza qui sono in cms. La `scale` viene utilizzata per le dimensioni del carattere.

Schema di base di un poster del proiettore

In orientamento orizzontale

```
\documentclass[final,t]{beamer}
\mode<presentation>
{
  \usetheme{Berlin}
}

\usepackage[orientation=landscape,size=a1,scale=1,debug]{beamerposter}
\usepackage{lipsum} % for dummy text

\title[]{\huge Awesome title}
\author[]{\large \textbf{Author Name1} \and Author Name2 \and Author Name3}
\institute[]{\Large Dept of XYZ, ABC Institute}
\date{}

\begin{document}

\begin{frame}
\maketitle
\begin{columns}[t]
  \begin{column}{.32\linewidth}

  \begin{block}{Some heading}
  \lipsum[1]
  \end{block}


```

```

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\end{column}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\begin{column}{.32\linewidth}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\end{column}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\begin{column}{.32\linewidth}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\end{column}
\end{columns}

\end{frame}

\end{document}

```

Some heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Some heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Some heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

<https://riptutorial.com/it/latex/topic/10714/creazione-di-poster-con-beamer>

Capitolo 10: Definizione di macro

Sintassi

- `\newcommand {\macro} {testo sostitutivo}`
- `\newcommand {\macro} [argomento] {testo sostitutivo}`
- `\renewcommand {\macro} {testo sostitutivo}`
- `\renewcommand {\macro} [argomento] {testo sostitutivo}`

Parametri

Parametro	Dettagli
<code>\macro</code>	La macro da definire
<code>argcount</code>	Il numero di argomenti che la macro si aspetta (opzionale)
<code>replacement text</code>	Il testo sostitutivo per la macro. All'interno di quel testo #1 , #2 ecc. Vengono sostituiti gli argomenti macro.

Examples

Definizione di base di macro

Definire un nuovo comando di base

Una macro può essere definita usando `\newcommand` . Per esempio:

```
\newcommand{\foo}{Just foo, you see?}
```

definisce una macro `\foo` che si espande in `Just foo, you see?` . Può quindi essere usato come qualsiasi comando integrato, ad esempio dopo quella definizione:

```
He said: ``\foo''
```

si espande a

```
He said: ``Just foo, you see?''
```

Definire un nuovo comando con argomenti

Le macro possono anche avere argomenti. Il numero di argomenti viene fornito come argomento

facoltativo tra il nome del comando e il testo sostitutivo. Nel testo sostitutivo, gli argomenti sono accessibili con #1 , #2 ecc. Ad esempio:

```
\newcommand{\better}[2]{A #1 is better than a #2.}
\better{solution}{problem} % gives: A solution is better than a problem
```

Ridefinire un comando esistente

Se una macro è già stata definita, `\newcommand` restituisce un errore. Per dare una nuova definizione per un comando esistente, viene utilizzato `\renewcommand` . A parte il nome diverso, la sintassi è esattamente la stessa. Ad esempio, dopo la definizione di `\foo` sopra, si potrebbe usare:

```
\renewcommand{\foo}{Another foo, please.}
```

Dopo quella ridefinizione, la macro `\foo` non si espande più a `Just foo, you see?` ma ad un `Another foo, please.`

Leggi Definizione di macro online: <https://riptutorial.com/it/latex/topic/7658/definizione-di-macro>

Capitolo 11: Formattazione del testo

Examples

Emphazise Text

Per enfatizzare il testo, è possibile utilizzare il comando `\emph` che di solito visualizza il testo in un carattere corsivo:

```
This is some text with \emph{emphasized words}.
```

Colpisci il testo

Il comando `\sout` del pacchetto `ulem` colpisce un testo:

```
\sout{This text is striked through}
```

Il pacchetto `ulem` ridefinisce il comando `\emph`. Quando non vuoi avere questo comportamento puoi usare il pacchetto `ulem` con l'opzione `normalem`:

```
\usepackage[normalem]{ulem}
```

Testo grassetto

Per comporre il testo in grassetto, usa `\textbf`:

```
\textbf{This text is typeset in bold.}
```

Leggi Formattazione del testo online: <https://riptutorial.com/it/latex/topic/7245/formattazione-del-testo>

Capitolo 12: Grafici di disegno

Examples

TikZ - Layout manuale

Il pacchetto [TikZ](#) si presta molto bene per disegnare grafici.

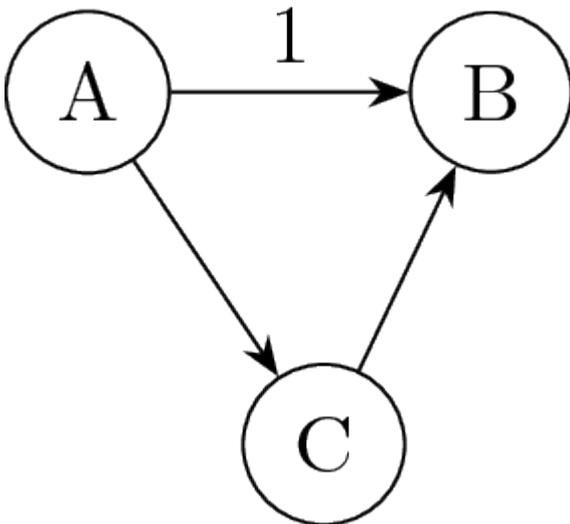
Questo è un piccolo esempio (richiede TikZ 3.0+):

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{positioning,arrows.meta}

\begin{document}
  \begin{tikzpicture}[auto,vertex/.style={draw,circle}]
    \node[vertex] (a) {A};
    \node[vertex,right=1cm of a] (b) {B};
    \node[vertex,below right=1cm and 0.5cm of a] (c) {C};

    \path[-{Stealth[]}]
      (a) edge node {1} (b)
      (a) edge (c)
      (c) edge (b);
  \end{tikzpicture}
\end{document}
```



Puoi creare grafici arbitrariamente complessi; fate attenzione al lungo codice, però. Ricorda che c'è `\foreach` e prendi nota di tutte le opzioni di posizionamento e stile (vedi il manuale di TikZ, sezione 13-17).

TikZ - Specifiche del grafico

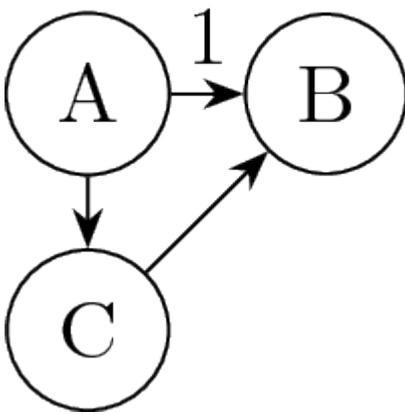
TikZ fornisce una sintassi simile a [DOT](#) che è possibile utilizzare per aumentare

considerevolmente il codice di disegno del grafico.

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{graphs,quotes,arrows.meta}

\begin{document}
  \begin{tikzpicture}
    \graph[nodes={draw,circle},edges={-{Stealth[]}}] {
      A -> ["1"] B,
      A -> C,
      C -> B
    };
  \end{tikzpicture}
\end{document}
```

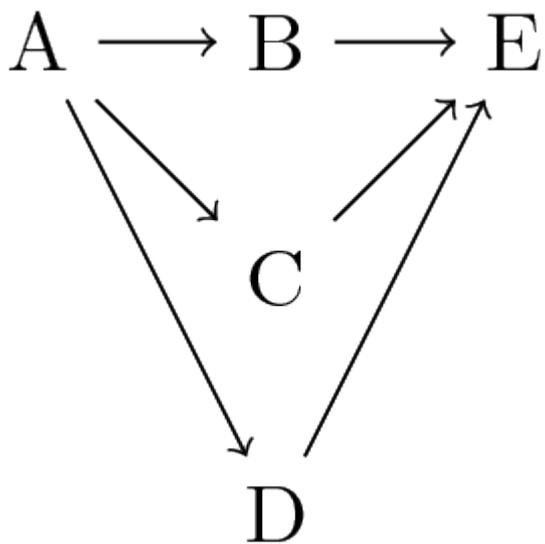


Come puoi vedere, scambi un controllo a grana fine per una sintassi più semplice. La libreria dei `graphs` brilla davvero quando specifichi grafici più complicati:

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{graphs,graphs.standard}

\begin{document}
  \begin{tikzpicture}
    \graph {
      A -> { subgraph I_n [V= {B,C,D}] } -> E
    };
  \end{tikzpicture}
\end{document}
```



Ci sono molte più opzioni e grafici predefiniti; vedere la sezione 19 del manuale TikZ.

TikZ - Disegno grafico algoritmico

TikZ implementa diversi algoritmi per *il* layout grafico *automatico* (richiede LuaLaTeX).

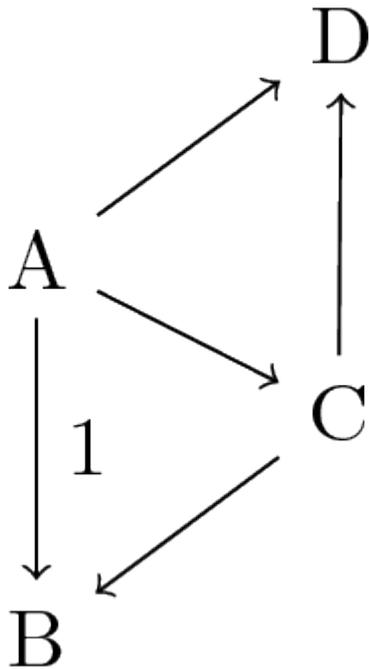
```

\documentclass{article}

\usepackage{tikz}
\usetikzlibrary{graphs,graphdrawing,quotes}
\usegdlibrary{force}

\begin{document}
  \begin{tikzpicture}
    \graph[spring layout] {
      A -> ["1"] B,
      A -> {C, D},
      C -> {B, D},
    };
  \end{tikzpicture}
\end{document}

```



Ci sono diversi algoritmi e molte opzioni per influenzarli. Vedere la parte IV del manuale di TikZ per i dettagli.

Diagramma di transizione di stato di una catena di Markov

Supponiamo che la seguente matrice sia la matrice di probabilità di transizione associata a una catena di Markov.

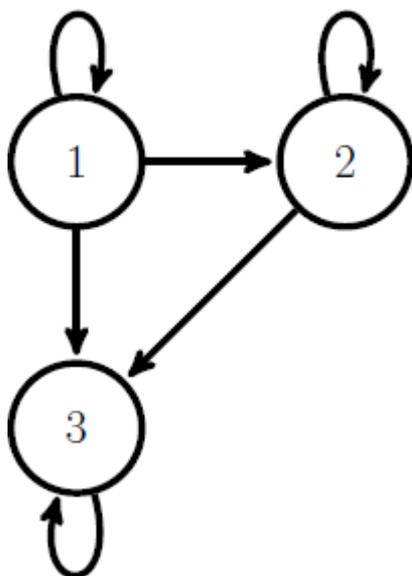
$$P = \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0.0 & 0.1 & 0.9 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$$

Per studiare la natura degli stati di una catena di Markov, viene disegnato un diagramma di transizione dello stato della catena di Markov.

```

\documentclass[12pt,a4paper]{article}
\usepackage{tikz}
\usetikzlibrary{shapes,arrows,positioning}
\begin{tikzpicture}[->,>=stealth',shorten >=2pt, line width=3pt,
                    node distance=2cm, style={minimum size=20mm}]
\tikzstyle{every node}=[font=\huge]
\node [circle, draw] (a) {1};
\path (a) edge [loop above] (a);
\node [circle, draw] (b) [right=of a] {2};
\path (b) edge [loop above] (b);
\draw[->] (a) -- (b);
\node [circle, draw] (c) [below=of a] {3};
\path (c) edge [loop below] (c);
\draw[->] (a) -- (c);
\draw[->] (b) -- (c);
  
```

```
\end{tikzpicture}
```



Leggi Grafici di disegno online: <https://riptutorial.com/it/latex/topic/5955/grafici-di-disegno>

Capitolo 13: Intestazione e piè di pagina

Examples

Usando i pacchetti fancyhdr e titoli

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

\usepackage{titleps}
\usepackage{fancyhdr}
\usepackage{graphicx}
\usepackage{lipsum} % for dummy text

\pagestyle{myheadings}
\pagestyle{fancy}
\fancyhf{}

\setlength{\headheight}{30pt}

\renewcommand{\headrulewidth}{4pt}
\renewcommand{\footrulewidth}{2pt}

\fancyhead[L]{\includegraphics[width=1cm]{example-image-a}}
\fancyhead[C]{}
\fancyhead[R]{\rightmark}
\fancyfoot[L]{ABC}
\fancyfoot[C]{\textcopyright xyz}
\fancyfoot[R]{\thepage}

\begin{document}

\section{First section}
\subsection{One}
\lipsum[1-3]
\subsection{Two}
\lipsum[4-6]

\end{document}
```

1 F

1.1 C

Lorem
vestibul
mauris.
magna.
tique se
Cras vi
ultrices
est, iacu
ces bibo
at, mol
risus. D
eget ord

Nam
auctor l
et, tellu
magna,
Suspend
natoque
Aliquan
cursus l

Null
at, tinc
ummy p
massa a
leo. Ma
cipit a,
lorem. S
Integer
in tellus
eu enim

Capitolo 14: Pagine del titolo

Osservazioni

`\title{<title>}` , `\author{<author>}` e `\date{<date>}` memorizzano internamente il contenuto.

`\maketitle` produce una pagina di titolo standard con i valori precedentemente definiti.

Examples

Titolo pagina report standard

```
\documentclass{report}

\begin{document}

\title{I want to be a Wombat}
\author{Carl Capybara}
\maketitle

\end{document}
```

Questo creerà un frontespizio senza altri contenuti:

I want to be a Wombat

Carl Capybara

July 26, 2016

Leggi Pagine del titolo online: <https://riptutorial.com/it/latex/topic/3010/pagine-del-titolo>

Capitolo 15: Presentazione con pacchetto beamer

Parametri

tema	AnnArbor
tema dei colori	seahorse

Osservazioni

Per altri temi e colori puoi visitare [qui](#)

Examples

Semplice diapositiva del titolo dell'autore

```
\documentclass{beamer}

\mode<presentation>

\usetheme{AnnArbor}

\usecolortheme{seahorse}

\title[Short topic]{Awesome long topic}

\author[Name]{Full name}

\institute[Institute short form]{Full name of institute}

\date{\today}

\begin{document}

\maketitle

\end{document}
```

Awesome loc

Full na

Full name of

July 1,

Capitolo 16: Spartiti per incisioni

Examples

LilyPond

L'incisore di notazione LilyPond può essere utilizzato con LaTeX tramite il comando `lilypond-book`. Per prima cosa creiamo un documento LaTeX (con estensione `.lytex`) per incorporare la nostra musica in:

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

\begin{document}

\begin{center}
  {\fontsize{24pt}{24pt}\textbf{Twa Corbies}}\
\end{center}

\begin{flushright}
  \textsc{Your Name}
\end{flushright}

% We don't need to require anything for this because lilypond-book will process it.
\lilypondfile{TwaCorbies.ly}
\end{document}
```

Quindi creiamo il nostro file LilyPond (`.ly`), incluso il file `lilypond-book-preamble.ly` (che LilyPond saprà come trovare):

```
\version "2.16.2"

\include "lilypond-book-preamble.ly"

voice = <<
  \relative c' {
    \tempo "con affetto"
    \clef bass
    \key e \minor
    \time 3/4

    a a b | c a a | g a2 |
    a4 a b | c2 ~ c8 a8 | a8 g a2 |
    \bar "|."
  }
  \addlyrics{
    As I was wal -- king all a -- lane
    I heard twa cor -- bies make a mane.
  }
>>

\score {
  <<
    \new Staff = "voice" {
      \voice
```

```
    }
  >>
  \layout { }
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 90 4)
    }
  }
}
```

per costruire, eseguiamo quindi il comando `lilypond-book` :

```
lilypond-book --include=mymusicsourcedirectory/ --pdf TwaCorbies.lytex
```

che genererà un PDF contenente la tua musica incisa LilyPond:

con affetto



As I

4



I heard

Capitolo 17: tabelle

Examples

L'ambiente tabulare

L'ambiente `tabular` è il modo più semplice per creare una tabella in LaTeX e non richiede altri pacchetti.

```
\begin{tabular}{|lcr|}  
  left aligned column & center column & right column \\  
  \hline  
  text & text & text \\  
  text & text & text \\  
\end{tabular}
```

left aligned column	center column	right column
text	text	text
text	text	text

Il parametro (`|lcr|` nell'esempio) è chiamato **specifica** della **tabella** e indica a LaTeX quante colonne ci sono e come dovrebbero essere formattate. Ogni lettera rappresenta una singola colonna. I valori possibili sono:

Personaggio	Senso
l	colonna allineata a sinistra
c	colonna centrata
r	colonna allineata a destra
p <code>{width}</code> eg <code>p{5cm}</code>	colonna di paragrafo con larghezza definita
(carattere pipe)	linea verticale
(2 tubi)	2 linee verticali

Le celle sono separate dal carattere `&`. Una riga è terminata da 2 barre rovesciate `\\`.

Le linee orizzontali possono essere inserite usando il comando `\hline`.

Le tabelle sono sempre formattate per essere sufficientemente ampie da includere tutto il contenuto. Se un tavolo è `overfull hbox` grande, LaTeX stamperà avvisi di `overfull hbox`. Le possibili soluzioni includono l'uso dello specificatore `p{width}` o di altri pacchetti come `tabularx`.

Una tabella con intestazione di colonna che si estende su più colonne può essere creata usando il comando `\multicolumn{cols}{pos}{text}` .

```
\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
&\multicolumn{3}{|c|}{Income Groups}\\
\cline{2-4}
City&Lower&Middle&Higher\\
\hline
City-1& 11 & 21 & 13\\
City-2& 21 & 31 & 41\\
\hline
\end{tabular}
\end{center}
```

City	Income Groups		
	Lower	Middle	Higher
City-1	11	21	13
City-2	21	31	41

Si noti che il comando `\multicolumn` ha tre argomenti obbligatori: il primo argomento specifica il numero di colonne su cui si estende l'intestazione; il secondo argomento specifica la posizione dell'intestazione (l, c, r) ; e il terzo argomento è il testo per la voce. Il comando `\cline{2-4}` specifica la colonna iniziale (qui, 2) e la colonna finale (qui, 4) su cui deve essere disegnata una linea.

Tavolo da colorare

Per rendere la tabella più leggibile, seguire i modi per colorarla:

1. Righe
2. colonne
3. Linee
4. cellule

Righe da colorare

Usa `\rowcolor` (fornito da `colortbl` ; anche caricato da `xcolor` sotto l'opzione del pacchetto `[table]`). Esempio:

```
\documentclass{article}
\usepackage[table]{xcolor}

\begin{document}

\begin{tabular}{|l|l|l|l|}
\rowcolor{green}
A & B & C & \\
\rowcolor{red}
D & E & F & \\
\end{tabular}
```

```
G & H & I \\
\rowcolor{blue}
J & K & L
\end{tabular}

\end{document}
```

A	B	C
D	E	F
G	H	I
J	K	L

Colonne da colorare

Le colonne possono essere colorate usando i seguenti modi:

- Definire la proprietà del colore della colonna al di fuori del tag della tabella usando

`\newcolumntype` :

```
\newcolumntype{a}{>{\columncolor{yellow}}c}
```

- Definizione della proprietà del colore della colonna all'interno dei parametri della tabella

```
\begin{tabular}{ | >{\columncolor{red}} c | l | l }
```

Esempio:

```
\documentclass{article}
\usepackage[table]{xcolor}

\newcolumntype{a}{>{\columncolor{yellow}}c}
\newcolumntype{b}{>{\columncolor{green}}c}

\begin{document}

\begin{tabular}{ a | >{\columncolor{red}}c | l | b }
\hline
A & B & C & D \\
E & F & G & H \\
\hline
\end{tabular}

\end{document}
```

A	B	C	D
E	F	G	H

Linee da colorare

Usa `\arrayrulecolor` . Esempio:

```
\documentclass{article}
\usepackage[table]{xcolor}

\arrayrulecolor{blue}

\begin{document}

\begin{tabular}{| l | l | l | }
\hline
A & B & C \\
\hline
D & E & F \\
\hline
G & H & I \\
\hline
\end{tabular}

\end{document}
```

A	B	C
D	E	F
G	H	I

Celle da colorare

Usa `\cellcolor` . Esempio:

```
\documentclass{article}
\usepackage[table]{xcolor}

\begin{document}

\begin{tabular}{| l | l | l | }
\hline
A & B & C \\
\hline
D & E & \cellcolor{green}F \\
\hline
\end{tabular}
```

```
G & H & I \\
\hline
\end{tabular}

\end{document}
```

A	B	C
D	E	F
G	H	I

Possiamo anche definire i nostri colori usando il pacchetto `colortbl` . Di seguito sono riportati gli esempi di tag:

```
\definecolor{Gray}{gray}{0.85}
\columncolor[RGB]{230, 242, 255}
\columncolor[HTML]{AAACED}
```

Leggi tabelle online: <https://riptutorial.com/it/latex/topic/4956/tabelle>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con il lattice	Community , eyqs , Harry , hbaderts , jani , Louis , Nijin22 , Sean Allred , SnoringFrog , Spacedman , Sumner Evans , TuringTux , tversteeg
2	Accesso alla documentazione dei pacchetti LaTeX	Raphael , samcarter
3	Aggiungi citazione	Hamzawey
4	Classi di documenti	Community , heather
5	Composizione di matematica	heather
6	Contatori, se dichiarazioni e cicli con lattice	csekri
7	Costruisci strumenti	Sean Allred
8	Creare una bibliografia	Nijin22 , Raphael , Werner
9	Creazione di poster con beamer	Sukanya B
10	Definizione di macro	celtschk , strpeter
11	Formattazione del testo	celtschk , Stephan Kulla
12	Grafici di disegno	L.V.Rao , Raphael
13	Intestazione e piè di pagina	Sukanya B
14	Pagine del titolo	adn , Johannes_B , Nijin22 , Sam Whited
15	Presentazione con pacchetto beamer	Sukanya B
16	Spartiti per incisioni	Sam Whited

17	tabelle	L.V.Rao , Nijin22 , Nikita Jain , Werner
----	---------	--