Corso Intensivo di LaTeX

Seconda Giornata

Autore e speaker: Luca Tesei

Corso offerto dal Camerino Linux User Group

http:\\camelug.unicam.it

Costrutti LaTeX

- Comandi
- Dichiarazioni
- Ambienti (Environment)

Comandi

- In realtà tutti i costrutti LaTeX sono comandi
- Alcuni servono per costruire environment
- Altri hanno effetti diversi
- Intendiamo come comando in senso “stretto” qualcosa che produce un certo output
- Ad esempio \LaTeX o \TeX o \section

Sintassi dei comandi

- Un comando inizia sempre con un backslash: \
- Senza spazi intermedi deve seguire il nome del comando
- I nomi sono “case sensitive”: \LaTeX è diverso da \Latex
- Per nome del comando si intende la sequenza di caratteri che segue \ fino al primo carattere non lettera: spazio, carattere speciale ({, \, ;, ...), cifra, ...

Sintassi dei comandi

- Fanno eccezione i comandi formati da un solo carattere dopo il \
- Es: _ \' \` \@ ...

Sintassi dei comandi

- Un comando può avere degli argomenti (parametri)
  - Opzionali
  - Obbligatori
- Gli argomenti opzionali vanno tra parentesi quadre e vanno prima di quelli obbligatori
- Gli argomenti obbligatori vanno tra parentesi graffe
- Se non ci sono argomenti opzionali le parentesi quadre non vanno digitate

Sintassi dei comandi: esempi

- Comandi a zero argomenti: **`\LaTeX \@`**
- Comandi ad un argomento obbligatorio: **`\section {`*`titolo`*`}`**, **`\u{`*`stringa`*`}`**
- Comandi con argomenti opzionali e uno obbligatorio:

  `\usepackage [italian]{babel}`

  `\documentclass[12pt,twocolumn]{book}`
- Fra il nome del comando e gli argomenti possono comparire spazi

Alcuni comandi utili

- Per sottolineare una stringa:

  `\underline{stringa}`

- Per centrare una linea

  `\centerline{stringa}`

Dichiarazioni

- Una dichiarazione è un comando che non produce un particolare output
- Essa influenza l'output di una certa parte del documento
- Tale parte è chiamata “scope”, cioè campo d'azione
- Il campo d'azione è delimitato dai blocchi

Blocchi

- Un blocco è una parte di documento
  - Fra una coppia di parentesi graffe { ... } oppure
  - Fra una coppia **`\begin{nome}`** ... **`\end{nome}`**
- In un documento ci possono essere molti blocchi uno dentro l'altro
- In questo caso di parla di blocchi **annidati**
- Il blocco più esterno è quello delimitato da **`\begin{document}`** e **`\end{document}`**

Dichiarazioni

- Il campo di azione di una dichiarazione
  - Inizia nel punto in cui viene inserita
  - Termina alla fine del blocco più interno in cui è inserita
- Il comando `\em` che abbiamo visto è una dichiarazione
- Essa fa in modo che tutte le parole nel suo campo di azione siano enfatizzate

Campo d'azione

- Possiamo mettere la dichiarazione in qualsiasi punto di un blocco
- Quando si inizia un nuovo blocco tutte le impostazioni precedenti rimangono valide fino ad eventuali dichiarazioni nuove
- In genere creiamo un blocco apposito con le parentesi graffe e inseriamo la dichiarazione all'inizio del blocco
- In caso contrario il suo campo di azione parte dal carattere immediatamente successivo fino alla fine del blocco corrente

Campo d'azione: esempio

- Normalmente: `Il {\em Gatto con gli stivali} abita nella foresta del re.`
- Strano: `Il {Gatto con gli \em stivali} abita nella foresta del re.`
- Nel primo caso viene enfatizzato “Gatto con gli stivali”
- Nel secondo caso solo “stivali”

Dichiarazioni utili

- Per cambiare il tipo di font
  - Corsivo: \it
  - Grassetto: \bf
  - Sans serif (senza grazie): \sf
  - Slanted (tipo corsivo): \sl
  - Stampatello maiuscolo: \sc
  - Macchina da scrivere: \tt
  - Romanico con grazie (default): \rm

Dichiarazioni utili

- Per cambiare la larghezza del font (in ordine crescente):
  - \tiny \scriptsize \footnotesize \small
  - \normalsize (Default)
  - \large \Large \LARGE \huge \Huge

Environment

- Un environment inizia con

  `\begin{nome-environment}`

- Un environment termina con

  `\end{nome-environment}`

- Un environment costituisce un nuovo blocco interno al blocco corrente
- Tutto ciò che è contenuto nell'environment produce output influenzato dalla definizione dell'environment stesso

Environment da dichiarazioni

- Ogni dichiarazione ha il suo corrispondente environment

```
\begin{em} testo enfatizzato
\end{em}
\begin{tt} testo macchina da
  scrivere \end{tt}
```

- ...

Liste

- Per produrre liste LaTeX mette a disposizione tre environment
  - Liste con bullets: **`itemize`**
  - Liste numerate: **`enumerate`**
  - Liste di descrizione: **`description`**
- Ogni elemento della lista è inserito tramite il comando **`\item`**
- Le liste possono essere annidate annidando i corrispondenti environment

Lista non numerata: esempio

```
Per ottenere un documento:

\begin{itemize}

\item Scrivere il file .tex

\item Compilare il file con
  \LaTeX\ sperando che non ci
  siano errori

\item Trasformare il .dvi nel
  formato desiderato

\end{itemize}
```

Liste descrittive

- L'environment **`description`** serve a dare definizioni
- Il comando **`\item`** in questo caso riceve un argomento opzionale contentente il testo da da descrivere

```
\item[Gatto] Felino domestico che essenzialmente mangia e dorme.
```

Verbatim

- L'environment verbatim trasforma momentaneamente il LaTeX in una macchina da scrivere
- Tutto ciò che viene digitato nel .tex dopo **`\begin{verbatim}`** viene riportato nell'output esattamente com'è
- Il font di default è **`\tt`**
- Si può scrivere di tutto dentro il verbatim, tranne la sequenza **`\end{verbatim}`** :-))

Poesie

- L'ambiente **`verse`** è utile per scrivere poesie
- Ogni paragrafo rappresenta una strofa
- All'interno di una strofa si può andare a capo con il comando `\\`

A capo

- Il comando `\\` funziona anche fuori dall'ambiente **`verse`**
- Serve ad andare a capo senza cambiare paragrafo
- `\\*` fa la stessa cosa ma impedisce che si inserisca in quel punto un fine pagina
- Per lasciare una riga bianca nell'output basta inserire un nuovo paragrafo contenente **solamente**: `\ \\`
- Lo spazio è necessario: il comando `\\` funziona solo se c'è qualcosa nella riga

Formule matematiche

- TeX e LaTeX sono nati in ambito scientifico
- Molta attenzione è stata concentrata sulla funzionalità di scrittura di formule matematiche
- Anche **molto** complesse
- Vediamo i principali modi di inserire formule
- Vediamo i principali comandi disponibili in una formula

Nel testo o in evidenza

- Una formula matematica può apparire:
- All'interno di un normale paragrafo di testo
  - “Inline formula”
- Centrata ed evidenziata (magari numerata) fra due paragrafi
  - “Displayed formula”

Ambienti matematici

- LaTeX mette a disposizione diversi ambienti che coprono i vari casi
- All'interno di tutti questi ambienti il modo di interpretare il testo è diverso
- Si dice che il testo è interpretato in “math mode”

In math mode

- Gli spazi vengono ignorati
- La stringa “e ci fa” viene interpretata come la moltiplicazione di quattro variabili chiamate e c i f a
- Il font in cui appaiono le lettere è specifico per le formule (una specie di corsivo, ma con peculiarità)

$$eci\,fa$$

Formule inline

- Ci sono tre ambienti equivalenti:
- `\begin{math} ... \end{math}`
- `$ ... $`
- `\( ... \)`
- Questi ultimi due non prevedono il begin e l'end, ma sono environment a tutti gli effetti

Formule in evidenza

- Qui abbiamo quattro ambienti:
  - `\begin{displaymath}...\end{displaymath}`
  - `\[ ... \]`
  - `$$ ... $$`
  - `\begin{equation}...\end{equation}`
- I primi tre non numerano la formula
- L'ultimo assegna un numero progressivo alle formule che può essere riferito nel testo: "`L'Equazione~5 dice che...`"

Apici e pedici

- Per le potenze si usa il comando ^:
- `$x^2$` genera x^2
- `$2^{x-1}$` genera 2^{x-1}
- `$2^{2^n}$` genera 2^{2^n}
- I pedici si mettono con il comando _:
- `$x_0$` genera x_0
- `$x_{i-1}$` genera x_{i-1}

Frazioni

- Il comando **frac** serve a fare frazioni
- Sintassi: `\frac{numeratore}{denominatore}`
- Naturalmente numeratore e denominatore possono contenere qualunque formula, contenente anche altri **`frac`** annidati

Radici

- Sintassi: `\sqrt[n]{formula}`
- L'argomento opzionale è un numero per fare radici terze, quarte, ennesime...
- Se non è specificato l'argomento opzionale la radice si intende quadrata

Puntolini

- Orizzontali
  - Bassi: `\ldots`
  - Centrali: `\cdots`
- Verticali: `\vdots`
- Diagonali: `\ddots`

Lettere greche

- Minuscole: backslash seguito dal nome:
  - \alpha
  - \beta
  - ...
- Tranne per omicron che si scrive come la o
- Maiuscole:
  - \Gamma
  - \Lambda
- Tranne quando la maiuscola si scrive come in alfabeto latino (es. alpha maiuscola è A)

Lettere calligrafiche

- Dichiarazione **`\cal`**
- Attenti al campo d'azione!
- **`${\cal f}(x)$`**
- Diversi packages danno lettere calligrafiche alternative: amssymb, amsfonts, euscript, ...
- Si sa, le lettere non bastano mai! :-))

Operatori binari

- `+ - / < >`
- Oltre a questi ce ne sono a bizzeffe!
- **`\leq`** “less or equal” <=
- **`\geq`** “greater or equal” >=
- **`\cap`** intersezione
- **`\cup`** unione
- **`\vee`** or
- **`\wedge`** and
- .....

“a primo”

- L'apice in math mode produce il tipico “primo”
- Due apici vengono stampati in maniera consona per indicare “secondo”
- Si possono inserire quanti apici si vogliono
- Dopo tre, comunque, il risultato è antiestetico

Spaziatura

- Gli operatori binari generano alla loro destra e sinistra degli spazi
- Anche se non sono scritti nel .tex
- Il + e il – sono considerati unari (+1, -2, ...) quando appaiono all'inizio di una formula
- In quel caso non c'è lo spazio extra

Spaziatura

- LaTeX ne sa più di noi sul modo giusto di spaziare gli elementi di una formula
- Ma se proprio non ci piace possiamo forzare la spaziatura con i seguenti comandi:
- `\,` piccolo spazio
- `\:` spazio medio
- `\;` spazio normale
- `\!` spazio normale **negativo** (torna indietro)

Funzioni tipo log

- Se dobbiamo scrivere logaritmo di 15
- `$log 15$` interpreta log come tre lettere l o g moltiplicate
- Non è l'effetto giusto
- Per questo ci sono dei comandi che generano i nomi delle funzioni più diffuse
- `$\log 15 + \sin \theta$`

Simboli a grandezza variabile

- Sommatorie (**`\sum`**), produttorie(**`\prod`**), unioni(**`\bigcup`**), intersezioni(**`\bigcap`**), integrali(**`\int`**), ...
- Somma dei primi n numeri naturali:
- **`$\sum_{i=1}^n i = \frac{n(n+1)}{2}$`**
- Si usano apice e pedice per specificare il range degli indici
- Attenzione: il modo in cui gli apici e i pedici vengono visualizzati è diverso nelle formule inline rispetto a quelle evidenziate

Array

- Come scrivere formule matematiche incolonnate e/o su più righe?
- L'environment **`array`** serve proprio a questo
- E' molto simile all'ambiente **`tabular`** che serve a fare tabelle (lo vedremo più avanti)
- Un array (o una tabella) consiste di n righe e m colonne.
- Il numero delle colonne va inserito come argomento dell'environment

Array

- `$\begin{array}{lcrc} ... \end{array}$`
- L'array ha 4 colonne
- La prima giustificata a sinistra (l di left)
- La seconda centrata (c di center)
- La terza giustificata a destra (r di right)
- La quarta centrata

Array

- Per separare gli elementi delle colonne di una stessa riga si usa il separatore `&`
- Se una casella è vuota il `&` va messo ugualmente:
- `... a + b & f(x) &  & 35 \\`
- Per separare le righe si usa il comando `\\`

Array esempio

```
\begin{array}{lcrc}
x + \log x & 23 & z & x-y \\
3 + \sin \theta & -4 & & w \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0
\end{array}
```

LR mode

- All'interno di tutti gli ambienti matematici il testo viene interpretato in math mode
- A volte c'è l'esigenza di scrivere una parola, ad esempio il nome di una funzione non standard oppure piccole particelle tipo se, o, e
- Il comando `\mbox{stringa}` può essere inserito sia in math mode che in modo di default (paragraph mode)
- Esso fa sì che la stringa argomento venga interpretata in LR mode

LR mode

- Una stringa interpretata in LR mode:
- Viene interpretata come una serie di parole lette da sinistra a destra (LR)

MA

- In output **non** vengono prodotte interruzioni di linea (non va a capo)
- Ogni spazio della stringa corrisponde a uno spazio in output (più spazi non vengono collassati in uno)

Parole in math mode

- Passare in LR mode può essere utile in math mode per scrivere parole
- In alternativa ci sono dei comandi per inserire parole in formule
- `\mathrm{stringa}` inserisce la stringa in font roman (di default)
- `\mathtt{stringa}` inserisce la stringa in font macchina da scrivere
- `\mathit{stringa}`
- ...

Delimitatori

- Ad esempio una matrice delimitata da due parentesi quadre grandi

```
\left [ \begin{array}{ccc}
          x & 4 & 7 \\
          0 & 0 & z \\
          z - x & 0 & 8
\end{array} \right ]
```

Delimitatori

- Simboli che agiscono come parentesi
- `( )  [ ]  \{ \}`
- Anche altri: **`\langle`**, **`\rangle`**, ...
- Possono essere allungati per delimitare formule di qualsiasi altezza
- Tramite i comandi **`\left`** e **`\right`**

Delimitatori

- A volte il delimitatore serve solo da una parte
- In ogni caso però `\left` e `\right` vanno messi in coppia
- Si può usare un . per dire che un certo delimitatore non è presente

Delimitatori

```
\[
f(x) = \left \{ \begin{array}{lll}
         0 & \mbox{ if } & x = 0\\
         x f(x-1) & \mbox{ if } & x > 0
         \end{array} \right .
\]
```

Cases

- Lo stesso risultato si può ottenere con

```
\[f(x) = \cases{
   0 & se $x=0$ \cr
   x f(x-1) & se $x>0$ \cr}\]
```

- Si noti che in ogni riga, dopo il &, ci si trova in LR mode
- Per inserire una formula bisogna rientrare in math mode

Allineamento di array

- A volte l'allineamento standard delle caselle di un array non produce l'effetto voluto
- L'ambiente array ha un parametro opzionale che può essere:
  - `t` = top La riga più in alto dell'array viene allineata con la linea ideale della riga
  - `b` = bottom La riga più un basso dell'array viene allineata con la linea ideale della riga

Allineamento degli array

```
\[ x - \begin{array}{c}
       a_1 \\ \vdots \\ a_n
       \end{array} -
   \begin{array}[t]{cl}
     u -v & 13 \\
     u + v & \begin{array}[b]{r}
                12 \\ -345
              \end{array}
   \end{array} \]
```

Linee verticali

- Per inserire linee verticali fra le colonne di un array basta inserire uno o più caratteri `|` fra le lettere che specificano l'allineamento della colonna
- `\begin{array}{||cr|c|l|} ...`
- L'array sarà delimitato a sinistra da due righe verticali
- Ci sarò una riga verticale tra la seconda, terza e quarta colonna.
- L'array sarà delimitato a sinistra da una linea verticale

Linee orizzontali

- All'inizio di una nuova riga o prima della prima riga si può inserire il comando **`\hline`**
- Esso inserisce una riga orizzontale sopra la riga di array corrente
- Per inserire una riga sotto all'ultima riga dell'array basta andare a capo nell'ultima riga e scrivere **`\hline`** come unico comando prima di **`\end{array}`**

Linee: esempio

```
\[ \begin{array}{|c||c||c|}
\hline
  \begin{array}{c}
    x=4 \\ x = + \infty
  \end{array}
& \mbox{ se } &
\begin{array}{c}
    x=5 \\ x = - \infty
  \end{array}\\
\hline
\end{array}\]
```

Spazi verticali e orizzontali

- `\hspace{15mm}` lascia uno spazio orizzontale di 15 millimetri
- `\vspace{1cm}` lascia uno spazio verticale di 1 centimetro
- Entrambi possono essere usati sia in math mode che in modo normale (paragraph mode)
- `\vspace` non funziona se inserito all'interno o all'inizio di una riga

Un simbolo sopra l'altro

- Alcune volte c'è la necessità di impilare simboli
- Ci sono alcuni comandi predefiniti in math mode
- **`\hat \bar \vec`** ... prendono un argomento e lo producono con un ^, una barra e la freccia (simbolo del vettore) rispettivamente
- **`\overline`** mette una riga sopra l'argomento

Un simbolo sopra un altro

- **`\overbrace`** pone una parentesi graffa orizzontale sopra il suo argomento
- **`\underbrace`** sotto
- **`\overbrace`** può prendere un superscript `^{...}` che va a finire sopra la parentesi graffa
- **`\underbrace`** fa la stessa cosa, sotto, con un subscript `_{...}`

Un simbolo sopra un altro

- Per creare nuove sovrapposizioni c'è il comando `\stackrel{arg1}{arg2}`
- arg1 viene posto sopra arg2
- arg1 ha dimensione superscript, mentre arg2 ha dimensione normale
- `\stackrel{a'}{\rightarrow}` è la classica freccia con a' sopra