

Pillole di L^AT_EX

Lezione IV: Beamer e Tikz

Angela Corvino
Gloria Cicconofri
Simone Ausilio
Giulio Carotta
Damiano Lucarelli

18 Dicembre 2020



Presentazioni con BEAMER

In Beamer, una presentazione consiste in una serie di quadri (in inglese *frame*), ciascuno dei quali è a sua volta costituito da una o più diapositive (*slide*).

- Tutto ciò che si trova tra `\begin{frame}` ed `\end{frame}` costituisce un quadro.
- Il comando `\frametitle` permette di assegnare un titolo al quadro corrispondente (per agevolare l'ascoltatore, è bene che ogni quadro abbia un titolo). Esiste anche il comando `\framesubtitle`, che consente di attribuire un sottotitolo al quadro corrispondente.

Il comando `\usetheme` serve per scegliere il tema della presentazione, ovvero l'aspetto generale delle diapositive quanto a colori e informazioni visualizzate. In questo caso si è scelto il tema `AnnArbor`, che è uno dei temi messi a disposizione da Beamer.

Blocchi

La classe Beamer mette a disposizione tre ambienti predefiniti per creare dei blocchi della diapositiva. Oltre a *block*, che si usa per blocchi generici, sono disponibili *exampleblock*, specifico per gli esempi, e *alertblock* per gli avvisi. Ogni tema di Beamer tratta i diversi blocchi in modo differente.

block

Questo è un esempio di blocco del tipo block.

exampleblock

Questo è un esempio di blocco del tipo exampleblock.

alertblock

Questo è un esempio di blocco del tipo alertblock.

Visualizzazione in più tempi di un elenco

```
\documentclass{beamer}
\title{La mia prima presentazione}
\author{Angela Corvino}
\date
\begin{document}
\maketitle
  \frametitle{Un esempio}
  \begin{itemize}
    \item<1-> Introduzione
    \item<2-> Corpo
    \item<3-> Conclusione
  \end{itemize}
```

Quella appena vista è una presentazione strutturata in due quadri: il primo, costituito da un'unica diapositiva, contiene il titolo della presentazione, mentre il secondo, suddiviso in tre diapositive, contiene un elenco.

- Il secondo quadro della presentazione è formato da tre diapositive.
Per scriverle nel codice, si usa un semplice ambiente `itemize`, nel quale ogni `\item` corrisponde a una diapositiva. Dopo ogni `\item` compaiono due parentesi all'interno delle quali sono indicate le diapositive da proiettare. La prima voce dell'elenco deve essere proiettata dalla diapositiva 1 del quadro in poi, il secondo elemento dalla diapositiva 2 in poi, eccetera;
- se le parentesi ad angolo contengono soltanto un numero, senza il trattino, la diapositiva viene proiettata solo quando tocca a lei, in base al numero d'ordine che la contraddistingue;

- l'assenza di un numero prima del trattino equivale a scrivere il numero 1; l'assenza di un numero dopo il trattino, invece, comporta che la voce dell'elenco verrà proiettata dalla diapositiva in cui compare per la prima volta fino all'ultima diapositiva del quadro.

Creare più colonne

Per creare colonne si usa l'ambiente `columns`; all'interno di questo ambiente si possono inserire diversi ambienti `column`, ognuno dei quali crea una nuova colonna.

```
\begin{frame}
\begin{columns}
  \begin{column}{0.4\textwidth}
    Mane \\ Tekel \\ Fares
  \end{column}
  \begin{column}{0.4\textwidth}
    Una riga (centrata).
  \end{column}
\end{columns}
```

Specificare la dimensione di ciascuna colonna è obbligatorio.

Per selezionare uno dei temi preimpostati (in questo caso "Madrid") basta inserire nel preludio:

```
\usetheme{Madrid}
```

Si possono anche combinare le caratteristiche di un tema con i colori di un altro (in questo caso "beaver"), basta aggiungere:

```
\usecolortheme{beaver}
```

Lista dei temi disponibili:

https://deic-web.uab.cat/~iblanes/beamer_gallery/index.html

Disegnare con TIKZ

Se vogliamo inserire dei disegni in un documento di LaTeX direttamente "programmandoli", un'opzione molto versatile è

```
\usepackage{tikz}
```

Adesso ogni volta che vogliamo disegnare qualcosa possiamo usare

```
\begin{tikzpicture}[ ... opzioni globali ... ]  
    ... istruzioni ...  
\end{tikzpicture}
```

Per piazzare meglio il disegno all'interno del documento possiamo anche racchiudere l'ambiente *tikzpicture* dentro un ambiente *figure*.

Griglia

Per definire ciascun punto del disegno Tikz utilizza una griglia di riferimento. Possiamo vederne una parte così:

```
\draw[help lines] (0,0) grid (2,2);
```

Abbiamo usato il comando *draw*, che ci permette di disegnare percorsi o figure più complesse. La sua sintassi è

```
\draw [opzioni] istruzioni ;
```

Occhio al ; in fondo a ogni *draw*! E' facilissimo dimenticarsene.

Coordinate

Ci sono due modi di esprimere coordinate in Tikz:

cartesiane:

(x , y)

polari:

$(\phi : \rho)$

Nel calcolo di una coordinata possiamo inserire anche qualche funzione di base e anche unità di misura (eventuali valori adimensionali vengono considerati espressi in "pt" se necessario).

Ad esempio

$(3\text{cm} + 2*\text{sqrt}(2) , 7\text{pt} - \sin(3*\text{pi } r))$

Disegnare una spezzata è molto facile. Iniziamo con un segmento:

```
\draw (0,0) -- (30:2);
```

Non serve usare tanti *draw*, possiamo continuare la linea così:

```
\draw (0,0) -- (1,1) -- (1,2) -- (0,2);
```

Posso anche andare accapo, il comando termina quando inserisco il punto e virgola. A volte è comodo usare coordinate relative rispetto al punto appena disegnato. Questo produce la stessa spezzata di prima:

```
\draw (0,0) -- ++(1,1) -- ++(0,1) -- ++(-1,0);
```

Rettangoli ed Ellissi

Tra due coordinate posso anche mettere altro rispetto a "--" (anche nulla!). Questo ci permette di disegnare figure più complicate, come abbiamo fatto per la griglia. Ad esempio, per disegnare un rettangolo specificando due vertici opposti:

```
\draw (0,0) -- (3,2);
```

Possiamo fare un cerchio così:

```
\draw ( posizione centro ) ellipse (raggio);
```

O un'ellisse:

```
\draw ( posizione centro ) ellipse (asse_x and asse_y);
```

Posso anche riempire le figure che faccio usando *fill* invece di *draw*. La sintassi è la stessa, ma i percorsi aperti verranno chiusi automaticamente.

Archi

La parola chiave *arc* si usa per fare archi di circonferenza. La sintassi è questa:

```
\draw ( punto di inizio ) arc  
      ( angolo partenza : angolo arrivo : raggio );
```

Se invece del punto iniziale conosco il centro posso usare una sintassi di questo tipo:

```
\draw (1,1) ++(45:7mm) arc (45:225:7mm);
```

Provate a ragionare su questo comando e su perché funzioni...

Per passare ai circuiti serve un ultimo comando:

```
\draw ( coordinate ) node [ opzioni ] { testo };
```

In questo modo possiamo inserire il *testo* centrato in *coordinate*. Questa è solo una delle possibili sintassi, ma ha il vantaggio di poter essere inserita dentro i percorsi che abbiamo fatto finora.

Un esempio più completo:

```
\draw [line width=10pt, blue] (0,0) -- (2,2)  
      node [align=right, draw] {A};
```

Circuiti con CIRCUITIKZ

Opzioni globali

Usare *tikz* per disegnare circuiti componente per componente è infattibile.
Fortunatamente viene in nostro aiuto *circuitikz*

```
\usepackage [opzioni]{circuitikz}
...
\begin{circuitikz}
    ... cose da disegnare ...
\end{circuitikz}
```

Questa volta è bene specificare almeno un'opzione: *american* o *european*, che indicherà qual'è lo stile grafico dei componenti circuitali che compariranno nel disegno. Altre opzioni possibili sono *americanvoltages*, *europeanvoltages*, *americaninductors*, *europeanresistors*, *emptydiodes*, ... che permettono di personalizzare ciascun componente singolarmente.

Corto Circuito

La cosa più semplice da disegnare è un corto circuito. Possiamo farlo con un semplice segmento di *tikz*

```
\draw (0,0) -- (2,0);
```

ma è meglio abituarsi alla sintassi di *circuitikz*

```
\draw (0,0) to [short] (2,0);
```

Questo può sembrare inutile, ma ci permette di avere più libertà nella personalizzazione, ad esempio aggiungendo un testo o cambiando gli estremi:

```
\draw (0,0) to [short, *-o, l=corto] (2,0);
```































Resistenze, Condensatori, Induttori, Generatori...

Con la stessa esatta sintassi possiamo collegare due punti del circuito con quasi qualunque componente

```
\draw (0,0) to [C=$10$ nF] (2,0);  
\draw (0,1) to [R, l^=$100 \Omega$] (2,1);  
\draw (0,2) to [L, l_=${L_1=50 mH}$] (2,2);  
\draw (0,3) to [V] (2,3);  
\draw (0,4) to [battery1] (2,4);  
\draw (0,5) to [vsourcesin] (2,5);  
\draw (0,6) to [diode] (2,6);  
\draw (0,7) to [ammeter] (2,7);  
\draw (0,8) to [voltmeter] (2,8);
```

Posso persino connettere due punti con un circuito aperto usando *open*. Perché è utile secondo voi?

Altri componenti di tipo "percorso"

ammeter		voltmeter		short	
open		lamp		generic	
tgeneric		ageneric		fullgeneric	
tfullgeneric		memristor		american resistor	
vR		american potentiometer		european resistor	
european resistor		european potentiometer		varistor	
photoresistor		thermocouple		thermistor	
thermistor ntc		fuse		afuse	
battery		battery1		european voltage source	
american voltage source		european current source		american current source	

Posizionamento Label e Indicazione Differenze di Potenziale

Il posizionamento di label l e annotations a è gestito tramite gli operatori \wedge (su) e $_$ (giù). La posizione è relativa al verso del componente, cioè da quale polo è il suo inizio e quale è la fine.

```
\draw (0,0) to [R, l^=$100 \Omega$] (2,0);  
\draw (0,1) to [R, l_=$100 \Omega$] (2,1);  
\draw (2,2) to [R, l_=$100 \Omega$] (0,2);  
\draw (0,3) to [R=$100 \Omega$] (2,3);  
\draw (0,3) to [R, l=$R_1$, a=10 $\Omega$] (2,3);
```


Indicazione Differenze di Potenziale

Le differenze di potenziale si indicano con v . Il risultato dipende molto dall'impostazione grafica scelta.

```
\draw (0,0) to [battery1, v^=3.3 V] (2,0);  
\draw (0,1) to [V=5 V] (2,1);  
\draw (0,2) to [open, v_=$V_{in}$] (2,2);
```

```
\begin{circuitikz}[american voltages]  
  \draw (0,0) to [battery1, v=3.3 V] (2,0);  
\end{circuitikz}  
\begin{circuitikz}[european voltages]  
  \draw (0,0) to [battery1, v=3.3 V] (2,0);  
\end{circuitikz}
```

Indicazione Correnti

Le correnti si possono indicare con i , che produrrà una freccia direttamente sul filo a un'estremità del componente, oppure con f (flow) che metterà la freccia a una certa distanza.

```
\draw (0,0) to [L, i^=$i_1$] (2,0);  
\draw (0,1) to [L, f_=$i_1$] (2,1);
```

Questa volta ci sono più posizionamenti possibili, gestiti dai 4 operatori \wedge , $_$, $<$ e $>$. Gli ultimi 2 indicano il verso della corrente, mentre l'ordine degli operatori (prima posizione e poi verso o viceversa) cambia la posizione della freccia rispetto al componente. Con degli esempi diventa molto più facile da capire:

```
\draw (0,0) to [R, i^>=$i_1$] (2,0);  
\draw (0,1) to [R, i_<=$i_2$] (2,1);  
\draw (0,2) to [R, f>_=$i_3$] (2,2);  
\draw (0,3) to [R, f<^=$i_4$] (2,3);
```

Nodi

E la terra? *ground* è un componente di tipo "nodo" (infatti ha un polo solo) e quindi si usa con la sintassi che già conosciamo per i nodi:

```
\draw (1,1) node [ground] {};
```

Tra le parentesi graffe è possibile mettere testo, che verrà sovrapposto al componente (se non specificata un'altra posizione) e quindi non è consigliato.









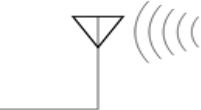
I nodi sono molto utili anche per marcare intersezioni tra fili:

```
\draw (0,0) to ++(2,0) (1,1) to --(0,2) (1,0) node [circ] {};
```

oppure per dare un nome a un punto del circuito:

```
\draw (0,0) to[short, -o] ++(1,0) node[above]{A} to[short] ++(0,1);
```

Altri componenti di tipo "nodo"

ground 	sground 	nground 
rground 	pground 	cground 
tlinestub 	antenna 	rxantenna 

Ruotare e Specchiare

I componenti di tipo "nodo" possono essere facilmente ruotati o specchiati (occhio che eventuali scritte vanno "anti-specchiate" a parte):

```
\draw (0,3) node[ground]{};
\draw (3,3) node[ground, rotate=90]{};
\draw (0,0) node[ground, rotate=180]{};
\draw (3,0) node[ground, rotate=270]{};
```

```
\draw (0,3) node[op amp]{OA1};
\draw (3,3) node[op amp, xscale=-1]{\scalebox{-1}[1]{OA2}};
\draw (0,0) node[op amp, yscale=-1]{\scalebox{1}[-1]{OA3}};
\draw (3,0) node[op amp, xscale=-1, yscale=-1]{\scalebox{-1}[-1]{OA4}};
```

Quelli di tipo "percorso" non hanno bisogno di queste opzioni aggiuntive, basta scambiarne gli estremi!

Componenti più Complicati

Nella slide precedente abbiamo usato come esempio un *op amp*, un componente a 5 poli. Questo è un componente di tipo "nodo", per cui non è ovvio come collegare i fili alle varie entrate/uscite. Possiamo però dare un nome al componente e poi richiamarne alcune sue parti predefinite come se fossero dei punti di ancoraggio (come le coordinate) per il circuito circostante.

```
\draw (0,3) node[op amp](OA1){};  
\draw (OA1.+) to [short, -*] ++(-1,0);  
\draw (OA1.-) to [short, -*] ++(-1,0);  
\draw (OA1.out) to [short, -*] ++(1,0);  
\draw (OA1.up) to [short, -*] ++(0,1);  
\draw (OA1.down) to [short, -*] ++(0,-1);
```

Ciascun componente ha le sue parti e i suoi nomi per richiamarle; perciò dobbiamo andare di volta in volta a vedere la sua documentazione specifica. In ogni caso non capiterà molto spesso...