

# TUTORIEL TikZ

## Introduction à TikZ

Karine Hébert  
Frédéric Quesnel



2023

# 1. TikZ

# Qu'est-ce que TikZ?

# Qu'est-ce que TikZ?

## PGF (*Portable Graphics Format*)

C'est un macro package  $\text{\LaTeX}$  qui permet la création d'éléments graphiques dans le corps du texte. Donc, il nous permet de dessiner des schémas comme si on était en train d'écrire du texte  $\text{\LaTeX}$ .

# Qu'est-ce que TikZ?

## PGF (*Portable Graphics Format*)

C'est un macro package  $\text{\LaTeX}$  qui permet la création d'éléments graphiques dans le corps du texte. Donc, il nous permet de dessiner des schémas comme si on était en train d'écrire du texte  $\text{\LaTeX}$ .

## TikZ

C'est la couche *user-friendly* de PGF pour faire des dessins en  $\text{\LaTeX}$ . Donc, on peut considérer que TikZ est une extension (package) de  $\text{\LaTeX}$

# Comment utiliser TikZ dans $\text{\LaTeX}$ ?

- TikZ vient déjà installé par défaut lorsque  $\text{\LaTeX}$  est installé sur le PC (MikTeX, TeXLive, MacTeX) ou lorsque l'on travaille sur Overleaf. Pour utiliser TikZ, on a qu'à ajouter la commande suivante dans le préambule du document :

```
\usepackage{tikz}
```

- Le document complet serait :

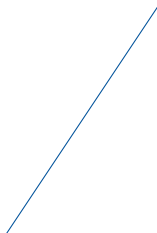
```
\documentclass{article}
\usepackage{tikz}

\begin{document}
  \begin{tikzpicture}
    commandes...
  \end{tikzpicture}
\end{document}
```

## 2. Concepts de base

# Un premier exemple avec TikZ

```
\begin{tikzpicture}  
\draw[options] (0,0) -- (2,3);  
\end{tikzpicture}
```



## Syntaxe:

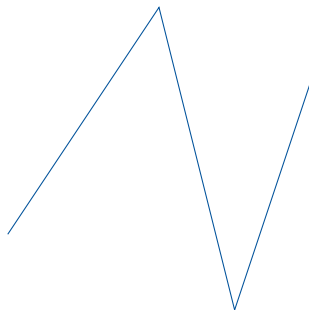
- `\draw` : fonction pour dessiner
- `[options]` : Options de ligne. Optionnel.
- `(0,0) -- (2,3)` : coordonnées des points. Par default, les coordonnées sont écrites en cm. Si l'on veut considerer d'autres systèmes, on doit le specifier `(0 in,0 in) -- (2 in,3 in)`
- `--` : l'opérateur d'extension
- `;` : fin de la commande !



# Opérateur d'extension

L'opérateur d'extension permet de concaténer une séquence de segments.

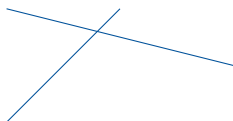
```
\begin{tikzpicture}  
\draw (0,0) -- (2,3) -- (3, -1) -- (4, 2);  
\end{tikzpicture}
```



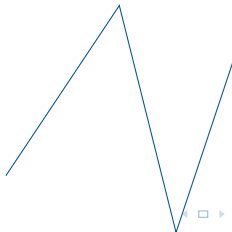
# Commande “;”

Permet d'écrire plusieurs commandes sur une seule ligne, ou permet d'écrire une seule commande sur plusieurs lignes :

```
\begin{tikzpicture}  
\draw (0,1) -- (2,3); \draw (0, 3) -- (4, 2);  
\end{tikzpicture}
```



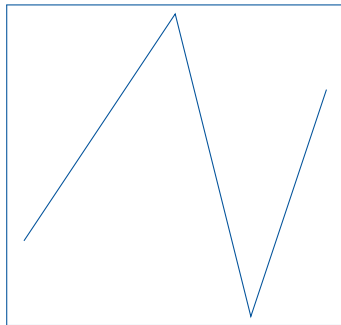
```
\begin{tikzpicture}  
\draw (0,0) --  
  (2,3) --  
  (3, -1) -- (4, 2);  
\end{tikzpicture}
```



# Placement des figures TikZ

Lorsque une figure est faite,  $\LaTeX$  la place dans un carré “imaginaire” qui englobe la figure dessinée!

```
\fbox{
\begin{tikzpicture}
\draw (0,0) --
(2,3) --
(3, -1) -- (4, 2);
\end{tikzpicture}
}
```



# Placement des figures TikZ

## Exemple

Soit la ligne

```
\begin{tikzpicture}  
\draw (0,0) -- (3,0);  
\end{tikzpicture}
```

, elle a été placée au milieu du texte.

Soit la ligne \_\_\_\_\_, elle a été placée au milieu du texte.

**Ou encore:**

Soit la ligne `\tikz \draw (0,0) -- (3,0);`, elle a été placée au milieu du texte.

Soit la ligne \_\_\_\_\_, elle a été placée au milieu du texte.

# Des figures TikZ ne sont pas des “figures” L<sup>A</sup>T<sub>E</sub>X!

La figure 1 illustre l'utilisation de TikZ comme figure L<sup>A</sup>T<sub>E</sub>X.

```
\begin{figure}[!htb]
\centering
\begin{tikzpicture}
\draw (0,0) -- (0,2) -- (2,2)
-- (2,0) -- (0,0);
\end{tikzpicture}
\caption{Le premier exemple
\TikZ}
\label{fig:exemple_tikz}
\end{figure}
```



Figure 1: Le premier exemple  
TikZ

- Ça permet à la figure TikZ d'avoir un comportement flottant!

## Remarque

Ce format est le plus utilisé dans la “vie réelle”!

# Redimensionner les figures TikZ

Prenons le carré dessiné précédemment:

```
\begin{figure}[!htb]
\centering
\begin{tikzpicture}
\draw (0,0) -- (0,2) -- (2,2)
-- (2,0) -- (0,0);
\end{tikzpicture}
\end{figure}
```



# Redimensionner les figures TikZ

Prenons le carré dessiné précédemment:

```
\begin{figure}[!htb]
\centering
\begin{tikzpicture}
\draw (0,0) -- (0,2) -- (2,2)
-- (2,0) -- (0,0);
\end{tikzpicture}
\end{figure}
```



Comment on peut faire un carré  $2\times$  plus grand et un carré  $2\times$  plus petit?



# Redimensionner les figures TikZ

```
\begin{tikzpicture}  
\draw (0,0) -- (0,4) --  
(4,4) -- (4,0) -- (0,0);  
\end{tikzpicture}
```

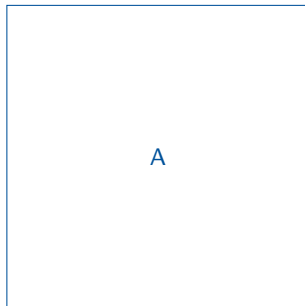


Figure 2: Carré 2× plus grand

```
\begin{tikzpicture}  
\draw (0,0) -- (0,1) --  
(1,1) -- (1,0) -- (0,0);  
\end{tikzpicture}
```



Figure 3: Carré 2× plus petit



# Redimensionner les figures TikZ

```
\begin{tikzpicture}  
\draw (0,0) -- (0,4) --  
(4,4) -- (4,0) -- (0,0);  
\end{tikzpicture}
```

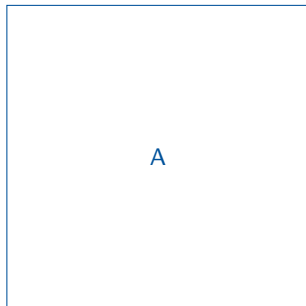


Figure 2: Carré 2× plus grand

```
\begin{tikzpicture}  
\draw (0,0) -- (0,1) --  
(1,1) -- (1,0) -- (0,0);  
\end{tikzpicture}
```

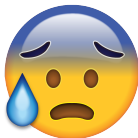


Figure 3: Carré 2× plus petit



# Redimensionner les figures TikZ

L'idée proposée avant marche bien, mais est-ce que cette approche est la plus pratique ? Et si on avait plusieurs coordonnées ?



# Redimensionner les figures TikZ

L'idée proposée avant marche bien, mais est-ce que cette approche est la plus pratique ? Et si on avait plusieurs coordonnées ?



Heureusement, il y existe une fonction qui peut rendre notre vie bien plus facile !



# Redimensionner les figures TikZ

```
\begin{tikzpicture}[scale=2]  
\draw (0,0) -- (0,2) -- (2,2)  
-- (2,0) -- (0,0);  
\end{tikzpicture}  
\caption{Carré 2X plus grand}
```

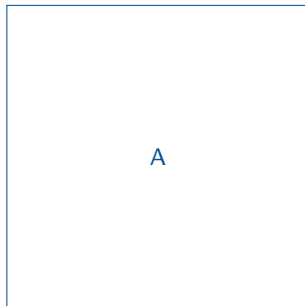


Figure 4: Carré 2× plus grand

```
\begin{tikzpicture}[scale=0.5]  
\draw (0,0) -- (0,2) -- (2,2)  
-- (2,0) -- (0,0);  
\end{tikzpicture}
```

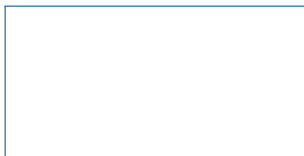


Figure 5: Carré 2× plus petit

# Redimensionner les figures TikZ

On a aussi la possibilité de redimensionner juste une des dimensions, c'est-à-dire, x et y.

```
\begin{tikzpicture}[xscale=2]
\draw (0,0) -- (0,2) -- (2,2)
-- (2,0) -- (0,0);
\end{tikzpicture}
```



```
\begin{tikzpicture}[yscale=2]
\draw (0,0) -- (0,2) -- (2,2)
-- (2,0) -- (0,0);
\end{tikzpicture}
```



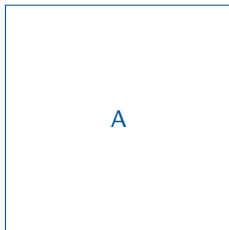
## Remarque

L'utilisation simultanée de `xscale` et `yscale` equivaut à `scale`!

# Redimensionner les figures TikZ (Bonus)

## resizebox

```
\begin{tikzpicture}[scale=1.5]
\draw (0,0) -- (0,2) --
      (2,2) -- (2,0) -- (0,0);
\end{tikzpicture}
```



```
\resizebox{3cm}{3cm}{
\begin{tikzpicture}
\draw (0,0) -- (0,1) --
      (1,1) -- (1,0) -- (0,0);
\end{tikzpicture}}
```



# Séparant le code TikZ du corps du texte

Mettre juste le code TikZ dans un fichier externe.

```
\begin{figure}[!htb]
\centering
\input{exemple_latex2}
\caption{Le premier exemple
\TikZ}
\label{fig:exemple_tikz3}
\end{figure}
```



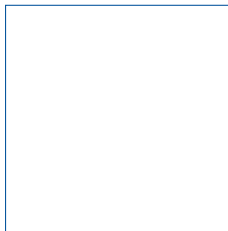
Figure 6: Le premier exemple  
TikZ

```
\begin{tikzpicture}
\draw (0,0) -- (0,2) -- (2,2) -- (2,0) -- (0,0);
\end{tikzpicture}
```

# Des formes géométriques basiques

On peut toujours dessiner des formes en jouant avec des segments.

```
\begin{tikzpicture}  
\draw (0,0) -- (0,3) --  
(3,3) -- (3,0) -- (0,0);  
\end{tikzpicture}
```





# Des formes géométriques basiques

On peut toujours dessiner des formes en jouant avec des segments.

```
\begin{tikzpicture}  
\draw (0,0) -- (0,3) --  
(3,3) -- (3,0) -- (0,0);  
\end{tikzpicture}
```



TikZ possède des commandes spéciales qui nous permettent de dessiner certaines formes plus facilement!

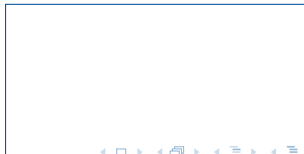
# Des formes géométriques basiques

## Rectangle

```
\begin{tikzpicture}  
\draw (0,0) rectangle (3,3);  
\end{tikzpicture}
```



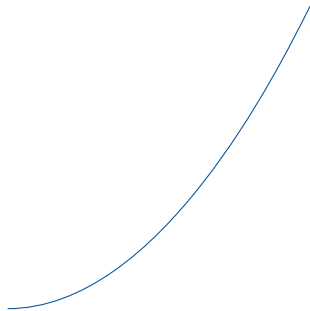
```
\begin{tikzpicture}  
\draw (0,0) rectangle (4,2);  
\end{tikzpicture}
```



# Des formes géométriques basiques

## Parabole

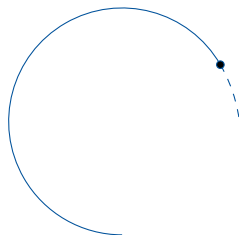
```
\begin{tikzpicture}  
\draw (0,0) parabola (4,4);  
\end{tikzpicture}
```



# Des formes géométriques basiques

## Arcs

```
\begin{tikzpicture}  
\draw (0,0) arc [radius = 1.5,  
start angle = 30,  
end angle = 270];  
\end{tikzpicture}
```



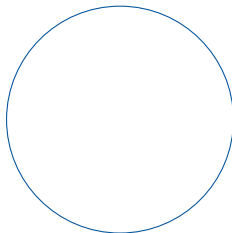
- (0,0) le point initial de l'arc
- arc commande pour dessiner l'arc
- radius
- start angle = 30 l'angle initial
- end angle = 270 l'angle final

**Remarque :** Les angles sont en degrés.

# Des formes géométriques basiques

## Arcs

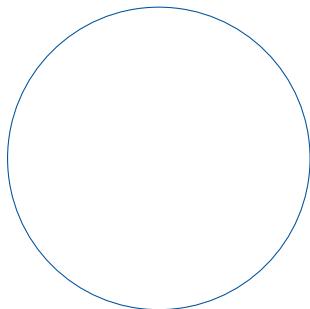
```
\begin{tikzpicture}  
\draw (0,0) arc [radius = 1.5,  
start angle = 0  
end angle = 360];  
\end{tikzpicture}
```



# Des formes géométriques basiques

## Cercle

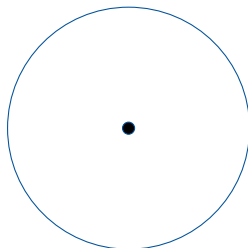
```
\begin{tikzpicture}  
\draw (0,0) circle (2cm);  
\end{tikzpicture}
```



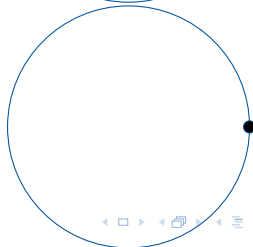
# Des formes géométriques basiques

## Cercle x Arcs

```
\begin{tikzpicture}  
\draw (0,0) circle (2cm);  
\end{tikzpicture}
```



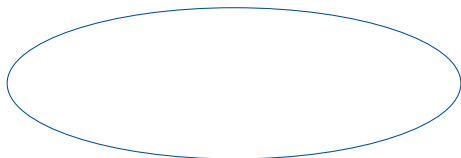
```
\begin{tikzpicture}  
\draw (0,0) arc [radius = 2,  
start angle = 0  
end angle = 360];  
\end{tikzpicture}
```



# Des formes géométriques basiques

## Ellipse

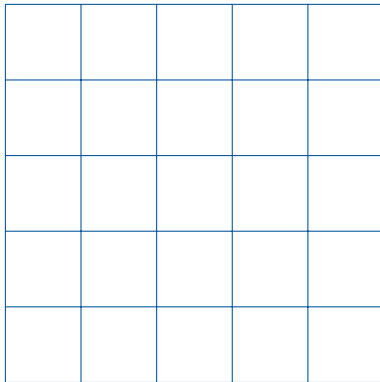
```
\begin{tikzpicture}  
\draw (0,0) ellipse (3cm and 1cm);  
\end{tikzpicture}
```





# Grilles

```
\begin{tikzpicture}  
\draw (0,0) grid (5,5);  
\end{tikzpicture}
```



# Décorations sur les lignes

- Lorsque on prépare des figures  $\text{\LaTeX}$ , on a besoin de les rendre plus “belles”
- La plupart des décorations peuvent être ajoutées en ajoutant des paramètres à la fonction `\draw!`

# Décorations sur les lignes

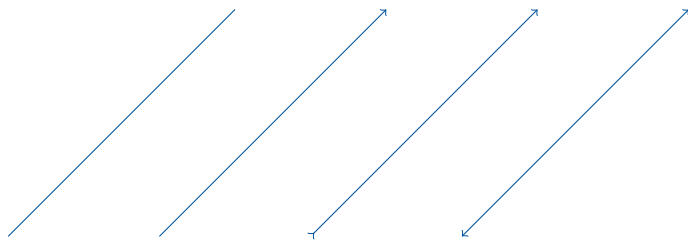
- Lorsque on prépare des figures  $\text{\LaTeX}$ , on a besoin de les rendre plus “belles”
- La plupart des décorations peuvent être ajoutées en ajoutant des paramètres à la fonction `\draw!`

## Décorations

- 1 Flèches et barres
- 2 Épaisseur des lignes (définie et personnalisée)
- 3 Lignes pointillées et lignes en tireté
- 4 Lignes courbées
- 5 Couleurs

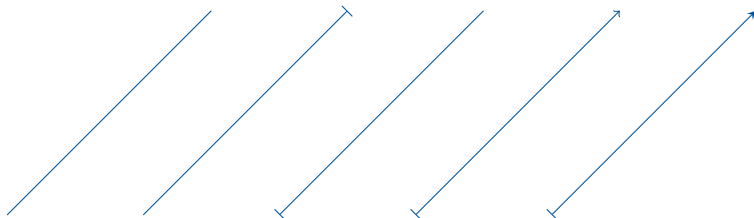
# 1. Flèches et barres

```
\draw (0,0) -- (3,3);  
\draw[->] (2,0) -- (5,3);  
\draw[>->] (4,0) -- (7,3);  
\draw[<->] (6,0) -- (9,3);
```



# 1. Flèches et barres

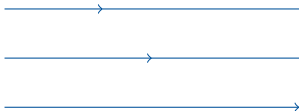
```
\draw (0,0) -- (3,3);  
\draw[-|] (2,0) -- (5,3);  
\draw[|-] (4,0) -- (7,3);  
\draw[|-|] (6,0) -- (9,3);  
\draw[|-stealth] (8,0) -- (11,3);
```



# 1. Flèches et barres

- Les flèches (barres) seront toujours ajoutées aux extrémités du premier et du dernier segment (si c'est le cas).
- TikZ (défaut) n'a pas une commande pour mettre des flèches au milieu des segments. On a besoin de "tricher"!

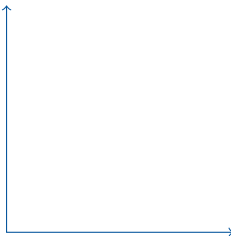
```
\begin{tikzpicture}  
\draw[->] (0,1) -- (1,1); \draw (1, 1) -- (3,1);  
\draw[->] (0,0.5) -- (1.5,0.5); \draw (1.5,0.5) -- (3,0.5);  
\draw[->] (0,0) -- (3,0);  
\end{tikzpicture}
```



# 1. Flèches et barres (Bonus)

C'est très bien pour faire des axes.

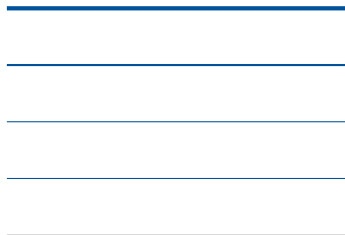
```
\begin{tikzpicture}  
\draw [<->] (0,3) -- (0,0) -- (3,0);  
\end{tikzpicture}
```



## 2. Épaisseur des lignes

### Définies

```
\draw[ultra thick] (0,2) -- (3,2);  
\draw[thick] (0,1.5) -- (3,1.5);  
\draw[thin] (0,1) -- (3,1);  
\draw (0,0.5) -- (3,0.5);  
\draw[help lines] (0,0) -- (3,0);
```



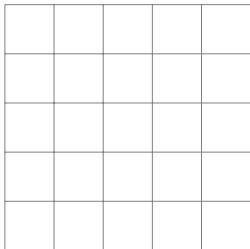


## 2. Épaisseur des lignes

### Définies - Lignes d'aide

Rendent les lignes plus “grises”, pâle et un peu plus étroites.

```
\begin{tikzpicture}  
\draw[help lines] (0,0) grid (5,5);  
\end{tikzpicture}
```

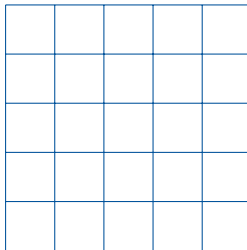
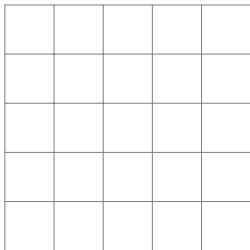


## 2. Épaisseur des lignes

### Définies - Lignes d'aide

Rendent les lignes plus “grises”, pâle et un peu plus étroites.

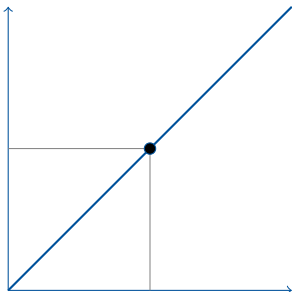
```
\begin{tikzpicture}  
\draw[help lines] (0,0) grid (5,5);  
\end{tikzpicture}
```



## 2. Épaisseur des lignes

Utiliser des “help lines” pour surligner des points.

```
\begin{tikzpicture}
\draw [<->] (0,5) -- (0,0) -- (5,0);
\draw[help lines] (0,2.5) -- (2.5,2.5) -- (2.5,0);
\draw[thick] (0,0) -- (5,5);
\draw[fill=black] (2.5,2.5) circle (0.1cm);
\end{tikzpicture}
```



## 2. Épaisseur des lignes

### Customisées

```
\draw[line width=12] (0,1) -- (3,1);  
\draw[line width=5] (0,0.5) -- (3,0.5);  
\draw (0,0) -- (3,0);
```



### Remarque

La dimension par défaut est le *point* (1 point  $\approx$  0.035 cm).

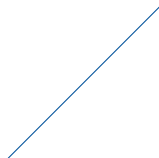
### 3. Lignes pointillées et lignes en tireté

```
\begin{tikzpicture}
\draw[dashed, ->] (0,2) -- (3,2);
\draw[dashed] (0,1.5) -- (3,1.5);
\draw[dotted] (0,1) -- (3,1);
\draw[dashed, ultra thick] (0,0.5) -- (3,0.5);
\draw[dashed, ->] (0,0) -- (3,0);
\end{tikzpicture}
```

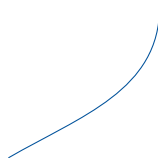


## 4. Lignes courbées

```
\begin{tikzpicture}  
\draw (0,0) -- (2,2);  
\end{tikzpicture}
```

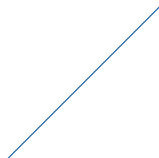


```
\begin{tikzpicture}  
\draw (0,0) to  
    [out=30,in=270] (2,2);  
\end{tikzpicture}
```

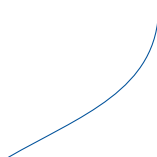


## 4. Lignes courbées

```
\begin{tikzpicture}  
\draw (0,0) -- (2,2);  
\end{tikzpicture}
```



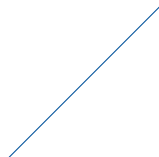
```
\begin{tikzpicture}  
\draw (0,0) to  
      [out=30,in=270] (2,2);  
\end{tikzpicture}
```



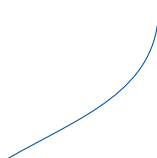
- **out**: Placer une aiguille avec une extrémité sur le point (0,0) et l'autre vers la droite, et ensuite la tourner dans le sens inverse des aiguilles d'une montre jusqu'à atteindre l'angle désiré.

## 4. Lignes courbées

```
\begin{tikzpicture}  
\draw (0,0) -- (2,2);  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
\draw (0,0) to  
      [out=30,in=270] (2,2);  
\end{tikzpicture}
```



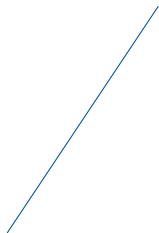
- **out**: Placer une aiguille avec une extrémité sur le point (0,0) et l'autre vers la droite, et ensuite la tourner dans le sens inverse des aiguilles d'une montre jusqu'à atteindre l'angle désiré.
- **in**: Placer une aiguille avec une extrémité sur le point (2,2) et l'autre vers la droite, et ensuite la tourner dans le sens inverse des aiguilles d'une montre jusqu'à atteindre l'angle désiré.



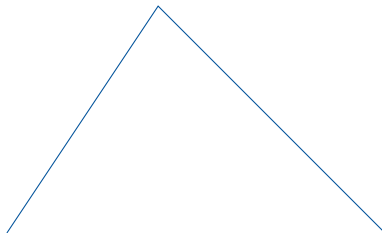
## 4. Lignes courbées

Prenez compte que l'opérateur `to` peut être utilisé pour dessiner des lignes droites. En fait, on peut même les mélanger.

```
\begin{tikzpicture}  
\draw (0,0) to (2,3);  
\end{tikzpicture}
```



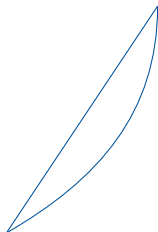
```
\begin{tikzpicture}  
\draw (0,0) -- (2,3) to (5,0);  
\end{tikzpicture}
```



## 4. Lignes courbées

Cependant, afin de rendre le code plus lisible et organisé, on va restreindre l'utilisation de "to" pour dessiner des courbes et "--" pour faire des lignes droites. C'est juste une bonne pratique.

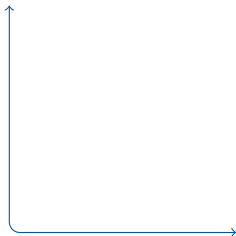
```
\begin{tikzpicture}  
\draw (0,0) -- (2,3);  
\draw (0,0) to [out=30, in=270] (2,3);  
\end{tikzpicture}
```



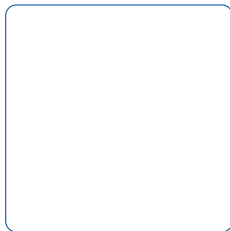
## 4. Lignes courbées

### Des coins ronds

```
\begin{tikzpicture}  
\draw [<->,  
rounded corners]  
(0,3) -- (0,0) -- (3,0);  
\end{tikzpicture}
```

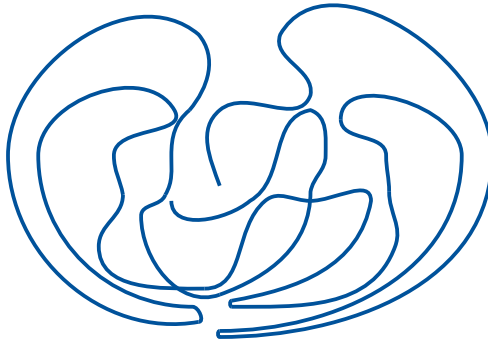


```
\begin{tikzpicture}  
\draw[rounded corners]  
(0,0) rectangle (3,3);  
\end{tikzpicture}
```



## 4. Lignes courbées

Un exemple un peu plus élaboré



## 4. Lignes courbées

### Un exemple un peu plus élaboré

```

\begin{tikzpicture}[scale=0.4]
\draw[line width=0.05cm] (5,2) to [out=110,in=220] (5,4.5) to [out=40,in=250]
(8,5) to [out=80,in=250] (7,7) to [out=60,in=90] (14,3) to [out=270,in=0]
(5,-3) to [out=180,in=180] (5,-2.8);
\draw[line width=0.05cm] (9,4.2) to [out=90,in=90]
(13,3) to [out=270,in=0] (5,-2.8);
\draw[line width=0.05cm] (10.6,1.8) to [out=90,in=270]
(10.5,3) to [out=90,in=270] (9,4.2);
\draw[line width=0.05cm] (10.6,1.8) to [out=270,in=70]
(10.8,0) to [out=250,in=0] (5.4,-2);
\draw[line width=0.05cm] (5.4,-1.8) to [out=180,in=180] (5.4,-2);
\draw[line width=0.05cm] (5.4,-1.8) to [out=10,in=260] (10,1.5) to [out=90,in=0]
(8,1.5) to [out=180,in=40] (6.5,1.5) to [out=220,in=0] (4.5,-1.4);
\draw[line width=0.05cm] (1.8,1.7) to [out=270,in=90]
(1,0) to [out=270,in=180] (4.5,-1.4);
\draw[line width=0.05cm] (1.8,1.7) to [out=100,in=270] (1.5,3) to [out=90,in=300]
(3.5,4.5) to [out=120,in=90] (-1,3) to [out=270,in=180]
(4.1,-2) to [out=0,in=0] (4.3,-2.5);
\draw[line width=0.05cm] (4,4.5) to [out=50,in=310] (4.3,7) to [out=140,in=90]
(-2,3) to [out=270,in=190] (4.3,-2.5);
\draw[line width=0.05cm] (3.2,1.7) to [out=60,in=270]
(3.4,3) to [out=90,in=220] (4,4.5);
\draw[line width=0.05cm] (3.2,1.7) to [out=250,in=90] (2,4,1);
\draw[line width=0.05cm] (5,-1.6) to [out=200,in=270] (2,4,1);
\draw[line width=0.05cm] (5,-1.6) to [out=20,in=270] (8,1.8) to [out=80,in=270]
(8.5,3) to [out=90,in=340] (8,4.5) to [out=190,in=30]
(6,1.5) to [out=220,in=270] (3,4,1.5);
\end{tikzpicture}

```

## 5. Couleurs

```
\begin{tikzpicture}  
\draw [line width = 6, red] (0,0) to (1,0);  
\end{tikzpicture}
```



Red



Black



Olive



Green



Gray



Orange



Blue



Darkgray



Pink



Cyan



Lightgray



Purple



Magenta



Brown



Teal



Yellow



Lime



Violete

Ou définissez vos propres couleurs!

```
\definecolor{BGERAD}{RGB}{0,82,155}
```

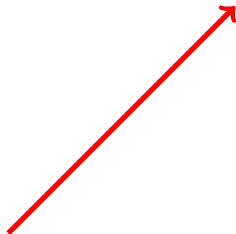


Bleu GERAD

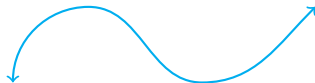
## 5. Couleurs

### Colorer des lignes

```
\begin{tikzpicture}
\draw [->,
      line width = 2,
      red] (0,0) -- (3,3);
\end{tikzpicture}
```



```
\begin{tikzpicture}
\draw [<->,
      thick, cyan]
(0,0) to [out=90,in=180]
(1,1) to [out=0,in=180]
(2.5,0) to [out=0,in=-135]
(4,1) ;
\end{tikzpicture}
```



## 5. Couleurs

### Colorer des figures bien définies

```
\begin{tikzpicture}  
\draw[fill=blue]  
  (0,0) rectangle (3,3);  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
\draw[fill=blue, red]  
  (0,0) rectangle (3,3);  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
\draw[fill=blue, blue]  
  (0,0) rectangle (3,3);  
\end{tikzpicture}
```

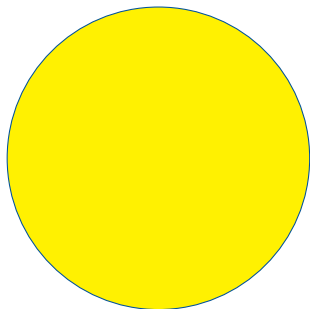




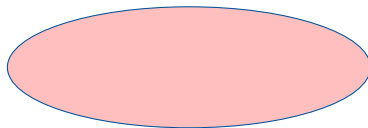
## 5. Couleurs

### Colorer des figures bien définies

```
\begin{tikzpicture}  
\draw[fill=yellow]  
(0,0) circle (2cm);  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
\draw[fill=pink]  
(0,0) ellipse (3cm and 1cm);  
\end{tikzpicture}
```



## 5. Couleurs

### Mélanger des couleurs

```
\begin{tikzpicture}  
\draw[fill=blue!30!red]  
  (0,0) rectangle (4,2);  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
\draw[fill=blue!50!red]  
  (0,0) rectangle (4,2);  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
\draw[fill=blue!70!red]  
  (0,0) rectangle (4,2);  
\end{tikzpicture}
```



#### Remarque

Ajouter `\usepackage{xcolor}` avant `\usepackage{tikz}`!

## 5. Couleurs

### Dégradé

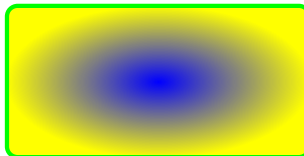
```
\begin{tikzpicture}  
\shade[top color=green,  
bottom color=yellow]  
(0,0) rectangle (4,2);  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
\shade[left color=green,  
right color=yellow]  
(0,0) rectangle (4,2);  
\end{tikzpicture}
```



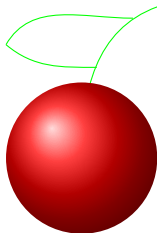
```
\begin{tikzpicture}  
\shadedraw[inner color=blue,  
outer color=yellow, draw=green]  
(0,0) rectangle (4,2);  
\end{tikzpicture}
```



## 5. Couleurs

### Dégradé

```
\begin{tikzpicture}  
\shade[ball color=red]  
  (0,0) circle (1cm);  
\end{tikzpicture}
```



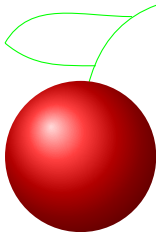
```
\begin{tikzpicture}  
\shade[ball color=green]  
  (0,0) rectangle (2,4);  
\end{tikzpicture}
```



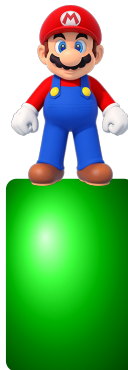
## 5. Couleurs

### Dégradé

```
\begin{tikzpicture}  
\shade[ball color=red]  
  (0,0) circle (1cm);  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
\shade[ball color=green]  
  (0,0) rectangle (2,4);  
\end{tikzpicture}
```



## 3. Graphes

## 3. Graphes

### Noeuds

# Créer des noeuds

$\backslash$ node [<options>] (<nom>) at (<X>,<Y>) {texte};

Options du noeud

Nom du noeud (optionnel)

Coordonnées

Texte (Obligatoire, peut être vide)

## Exemple:

```
 $\backslash$ node [] (monNoeud) at (0,0) {ceci est un noeud};
```

ceci est un noeud



# Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

# Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

2 [draw, circle]

# Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

2 [draw, circle]

3 [draw,  
rounded corners=3pt]

# Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

2 [draw, circle]

3 [draw,  
rounded corners=3pt]

4 [draw, red]

# Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

2 [draw, circle]

3 [draw,  
rounded corners=3pt]

4 [draw, red]

5 [draw, text=blue ]

# Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

6 [draw, fill=red!30]

2 [draw, circle]

3 [draw,  
rounded corners=3pt]

4 [draw, red]

5 [draw, text=blue ]

# Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

6 [draw, fill=red!30]

2 [draw, circle]

7 [draw, ultra thick]

3 [draw,  
rounded corners=3pt]

4 [draw, red]

5 [draw, text=blue ]

# Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

6 [draw, fill=red!30]

2 [draw, circle]

7 [draw, ultra thick]

3 [draw,  
rounded corners=3pt]

8 [draw, pattern =  
north east lines]

4 [draw, red]

5 [draw, text=blue ]



# Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

6 [draw, fill=red!30]

2 [draw, circle]

7 [draw, ultra thick]

3 [draw,  
rounded corners=3pt]

8 [draw, pattern =  
north east lines]

4 [draw, red]

9 [draw,  
inner sep=1pt]

5 [draw, text=blue ]

## Options de noeuds

```
\node [<options>] (<nom>) at (<X>,<Y>) {texte};
```

1 [draw]

6 [draw, fill=red!30]

2 [draw, circle]

7 [draw, ultra thick]

3 [draw,  
rounded corners=3pt]

8 [draw, pattern =  
north east lines]

4 [draw, red]

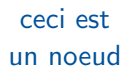
9 [draw,  
inner sep=1pt]

5 [draw, text=blue ]

10 [draw,  
minimum height = 30pt]

# Texte du noeud sur plusieurs lignes

```
\node [draw] (monNoeud) at (0,0) {  
  \begin{tabular}  
    ceci est \\ un noeud  
  \end{tabular}  
};
```



ceci est  
un noeud

## 3. Graphes

### Arcs

# Arcs

```
\draw[<options>] (<noeud 1>) --(<noeud 2>);
```

## Exemple :

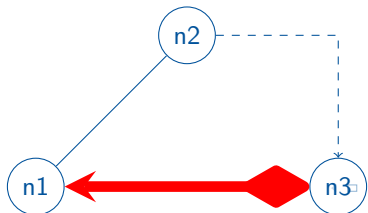
```
\node [draw, circle] (n1) at (0,0) {n1};
```

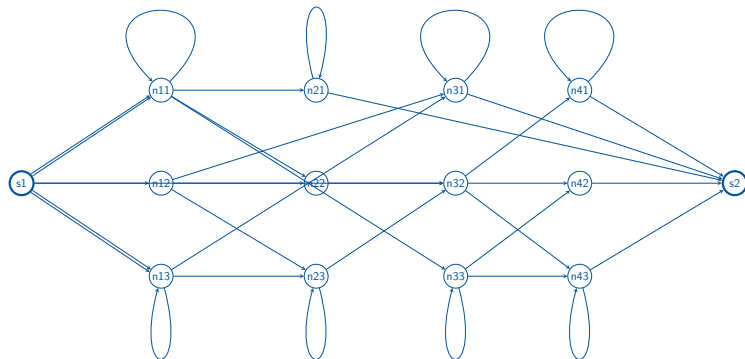
```
...
```

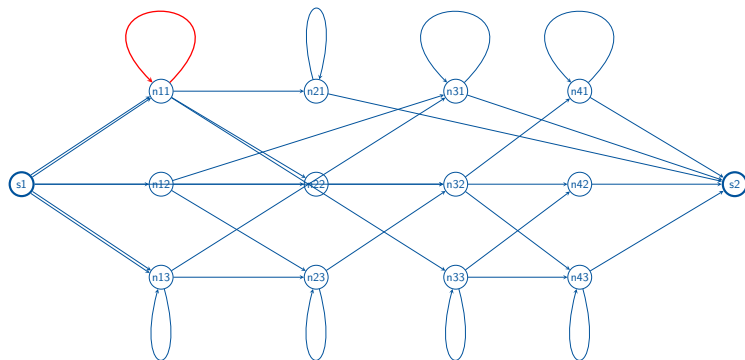
```
\draw[] (n1) -- (n2);
```

```
\draw[dashed, ->] (n2) -| (n3);
```

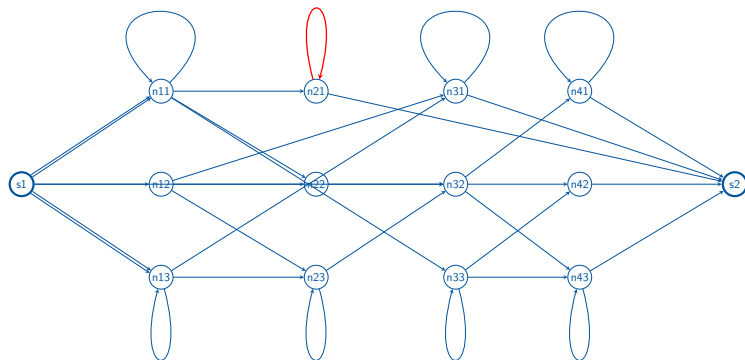
```
\draw[red, line width=4pt, diamond-stealth]  
      (n3) to (n1);
```





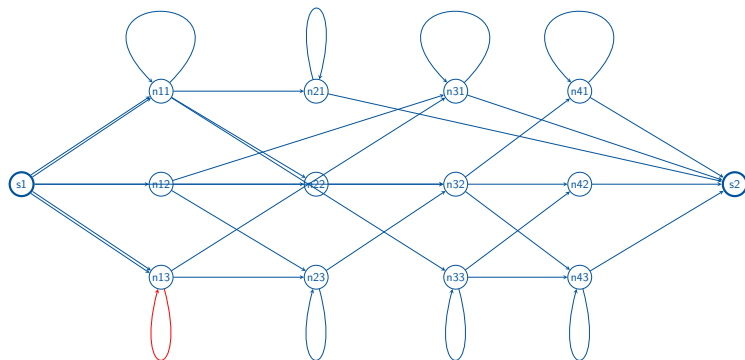


```
\draw[] (n11) to[loop, looseness=20] (n11);
```

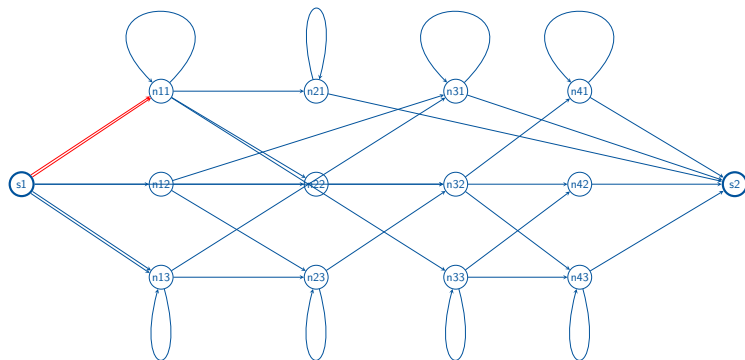


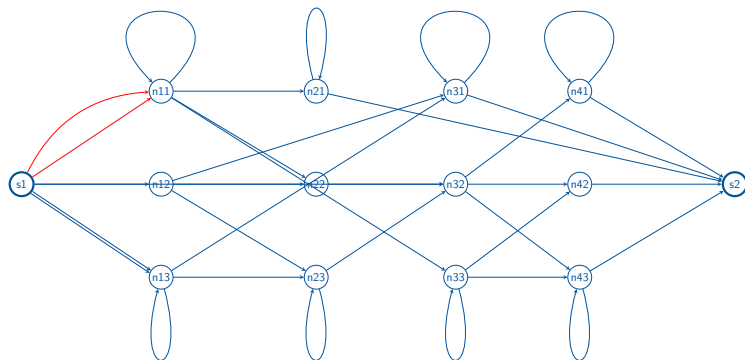
```
\draw[] (n21) to[loop above, looseness=40] (n21);
```



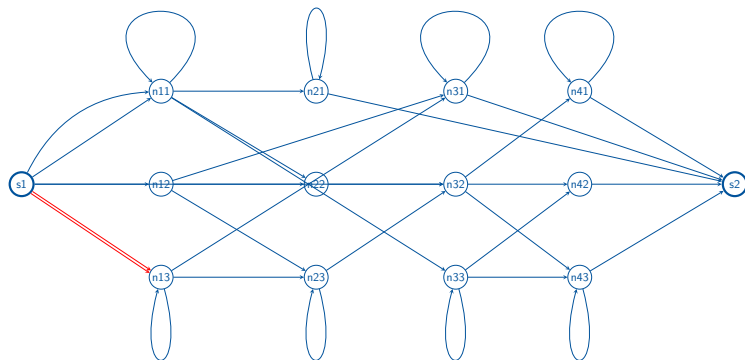


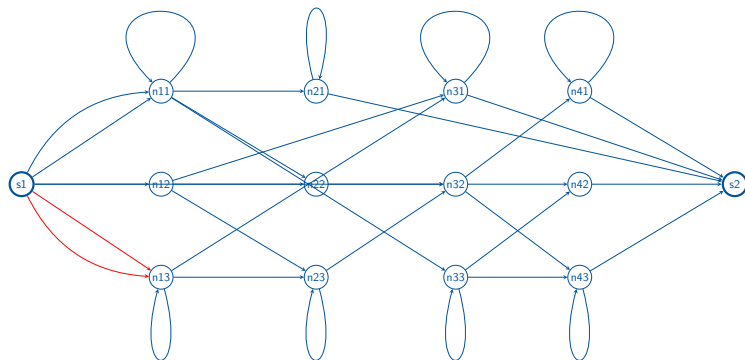
```
\draw[] (n13) to[loop below, looseness=40] (n13);
```



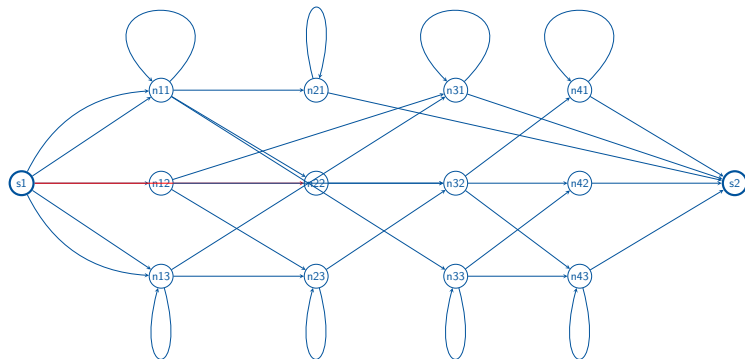


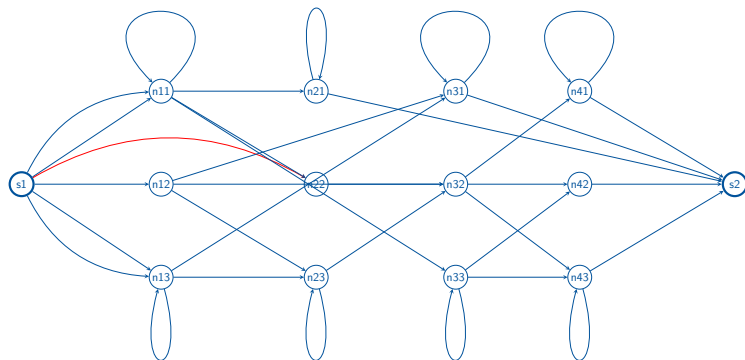
```
\draw[] (s1) to[bend left] (n11);
\draw[] (s1) --(n11);
```



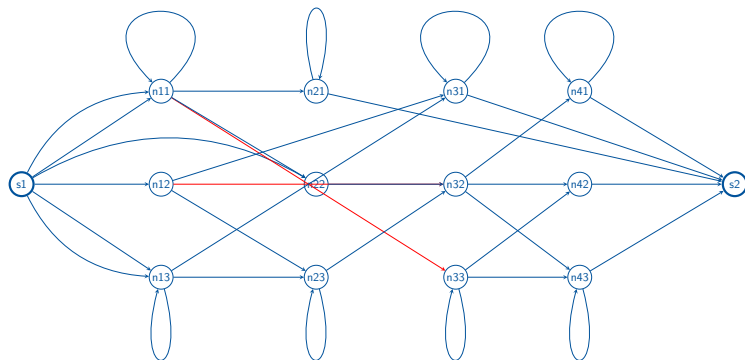


```
\draw[bend right] (s1) to (n13);  
\draw[] (s1) --(n13);
```

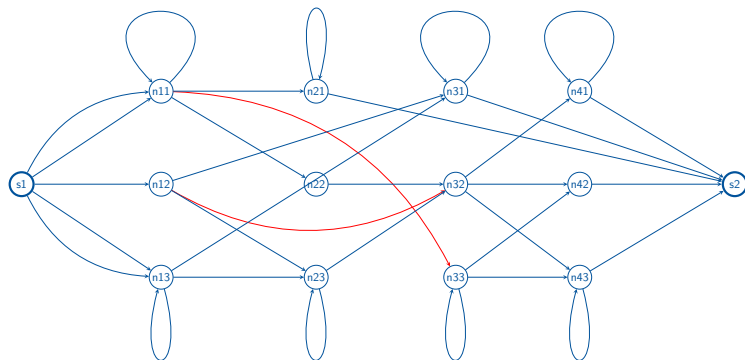


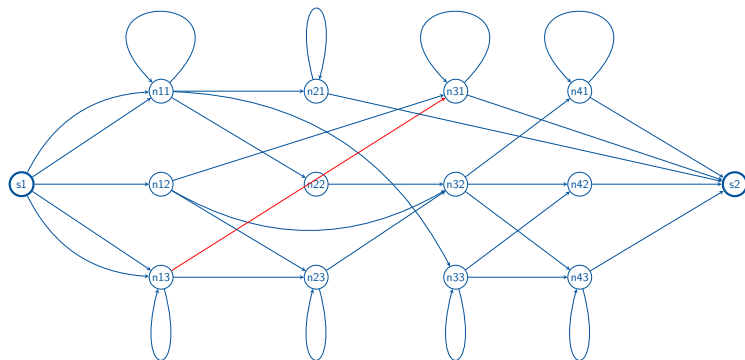


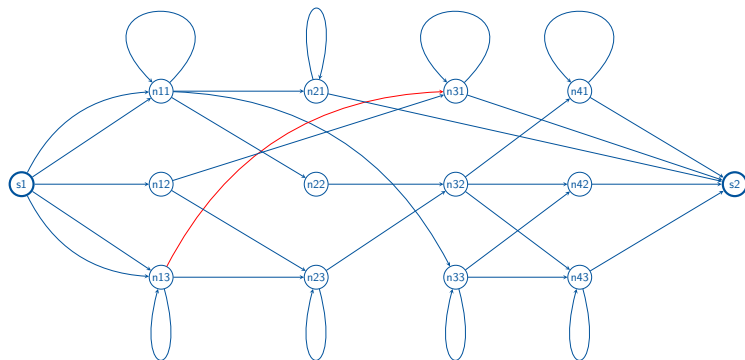
```
\draw[] (s1) to[bend left] (n22);
```



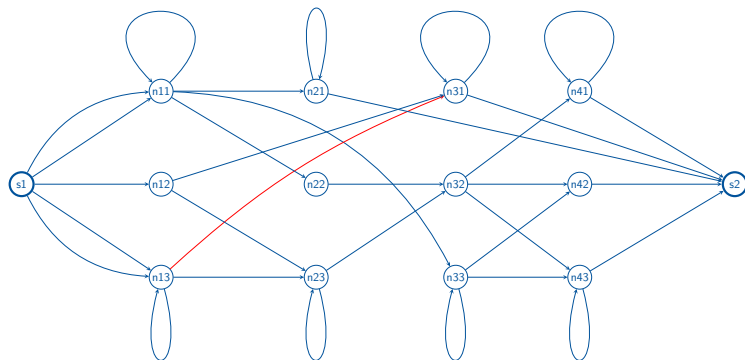




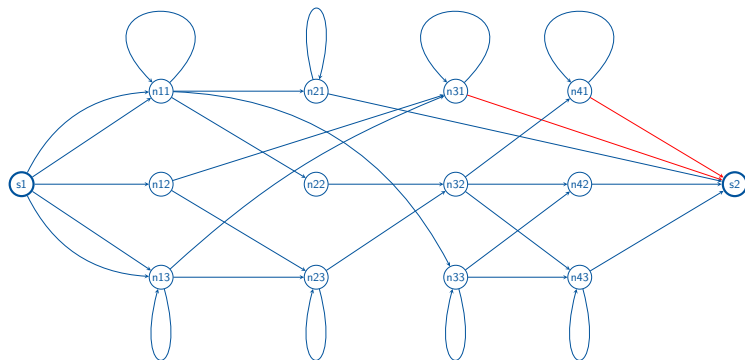


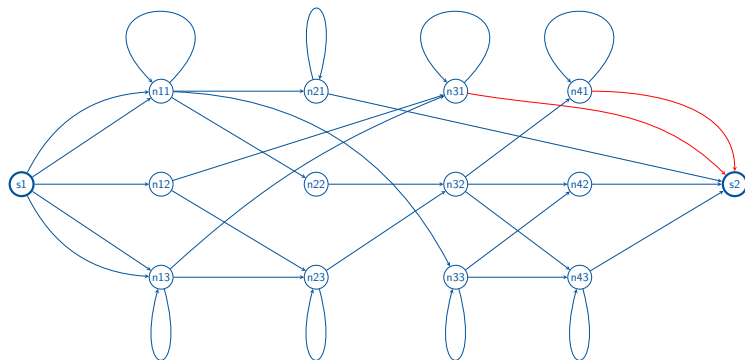


```
\draw[] (n13) to[bend left] (n31);
```

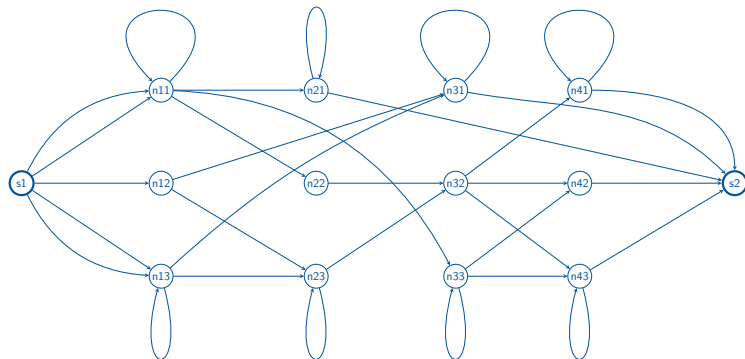


```
\draw[] (n13) to[bend left=10] (n31);
```





```
\draw[] (n31) to[out = -10, in = 135] (s2);  
\draw[] (n41) to[out=0, in=90] (s2);
```



## 3. Graphes

### Étiquettes



# Étiquettes de noeuds

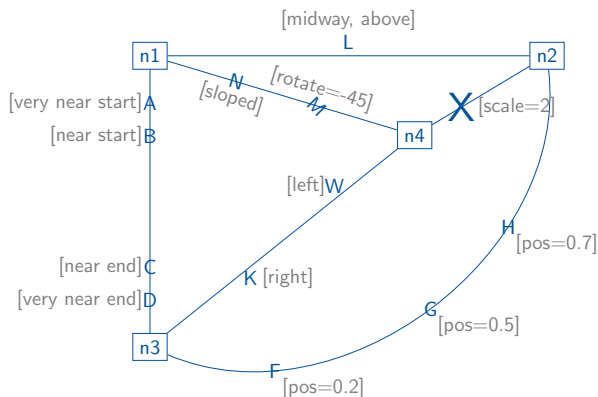
```
\node [draw, label = Étiquette du noeud n1] (n1) at (0,0) {};
```

```
\node [draw, label = {[red]45:n2}] (n2) at (2,0) {};
```

Étiquette du noeud n1



# Étiquettes d'arcs



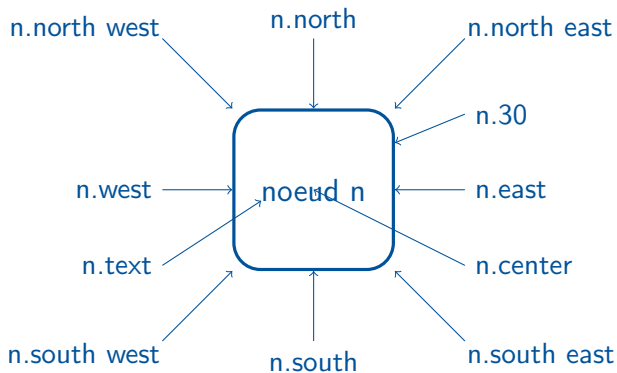
Avec la commande `--`

```
\draw [] (n1) --(n2) node[midway, above] {L};
```

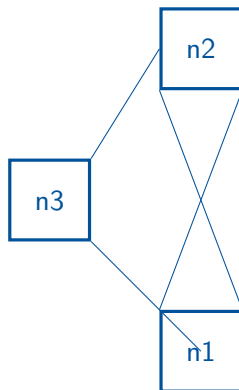
Avec la commande `to`

```
\draw[bend right = 60] (n3) to
node[pos=0.2] {F} node[pos=0.5] {G} node[pos=0.7] {H}
(n2);
```

# Ancres



# Ancres



```
\draw[] (n1.north west) --(n2.south east);  
\draw[] (n1.north east) --(n2.south west);  
\draw[] (n3.north east) --(n2.west);  
\draw[] (n3.south east) --(n1.center);
```

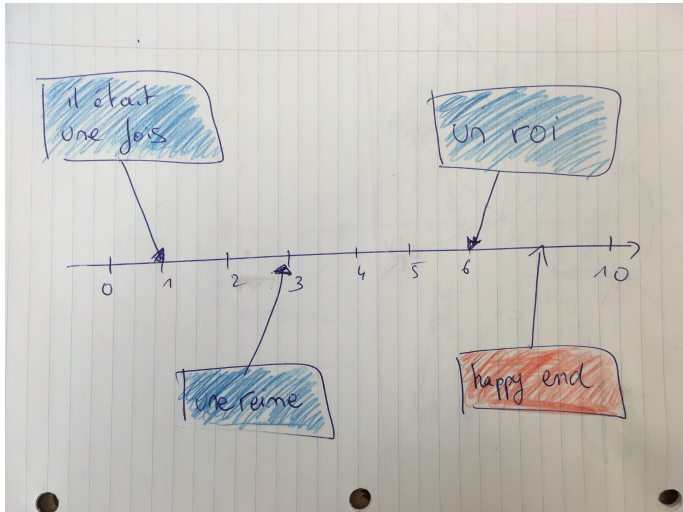
## 4. Vite fait, bien fait

# Vite fait, bien fait

Ou comment se simplifier la vie



# Exemple 1



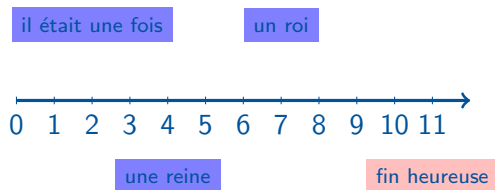
# Exemple 1



```
\draw[->,very thick] (0,0)--(12,0);  
\def\h{0.1}  
\foreach \i in {0,...,11}  
  \draw (\i,\h)--(\i,-\h) node[below] {\i};
```

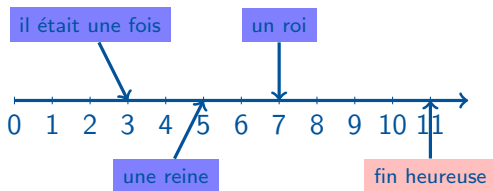


# Exemple 1



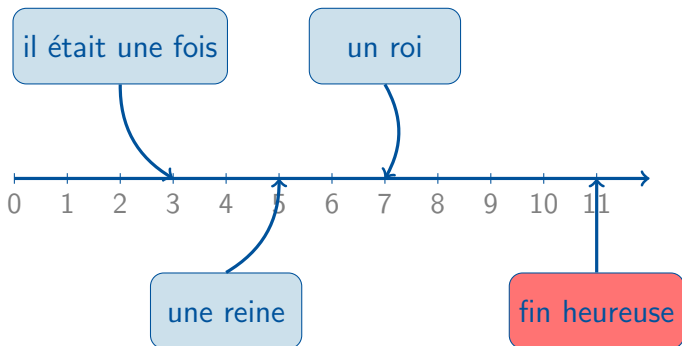
```
\def\z{2}
\tikzstyle{evenement} = [fill=blue!50,font=\large]
\node[evenement] (debut) at (2,\z) {il était une fois};
\node[evenement] (milieu) at (4,-\z) {une reine};
\node[evenement] (milieu2) at (7,\z) {un roi};
\node[evenement,fill=pink] (fin) at (11,-\z) {fin heureuse};
```

# Exemple 1



```
\tikzstyle{fleche} = [->,very thick]
\draw[fleche] (debut)--(3,0);
\draw[fleche] (milieu2)--(7,0);
\draw[fleche] (milieu)--(5,0);
\draw[fleche] (fin)--(11,0);
```

# Exemple 1: finitions

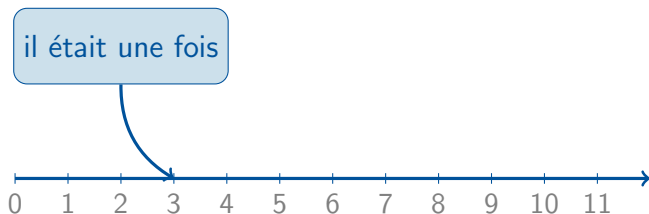


# Exemple 1: animations



Dans un beamer, utiliser les fonctions `\only` et `\uncover`.

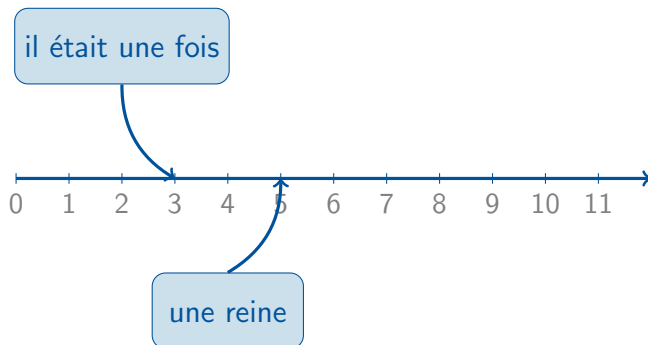
# Exemple 1: animations



Dans un beamer, utiliser les fonctions `\only` et `\uncover`.

```
\only<2->{ \node[event] (debut) at (4,-\z) {il etait une  
fois}; }
```

## Exemple 1: animations

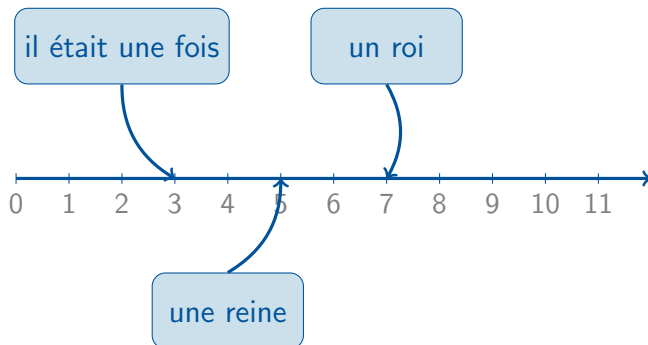


Dans un beamer, utiliser les fonctions `\only` et `\uncover`.

```
\only<2->{ \node[evenement] (debut) at (4,-\z) {il etait une  
fois}; }
```

```
\only<3->{ ... "une reine"... };
```

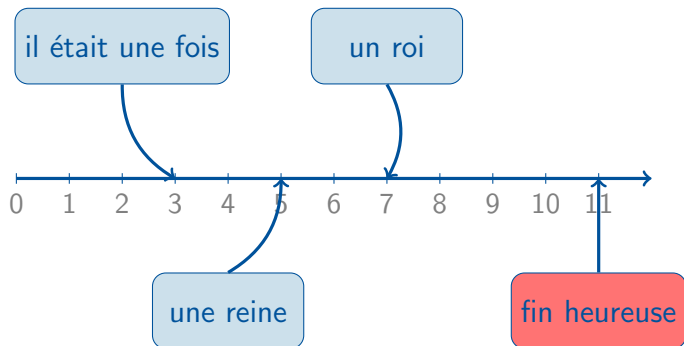
# Exemple 1: animations



Dans un beamer, utiliser les fonctions `\only` et `\uncover`.

```
\only<2->{ \node[evenement] (debut) at (4,-\z) {il etait une
fois}; }
\only<3->{ ... "une reine"... };
```

# Exemple 1: animations



Dans un beamer, utiliser les fonctions `\only` et `\uncover`.

```
\only<2->{ \node[evenement] (debut) at (4,-\z) {il etait une
fois}; }
\only<3->{ ... "une reine"... };
```

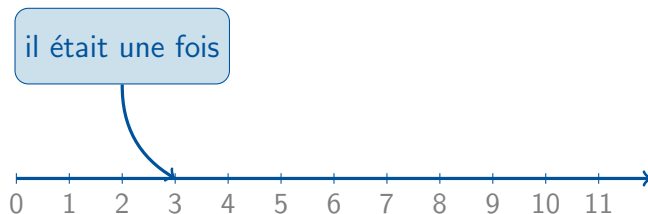


# Exemple 1: animations



Dans un beamer, utiliser les fonctions `\only` et `\uncover`.

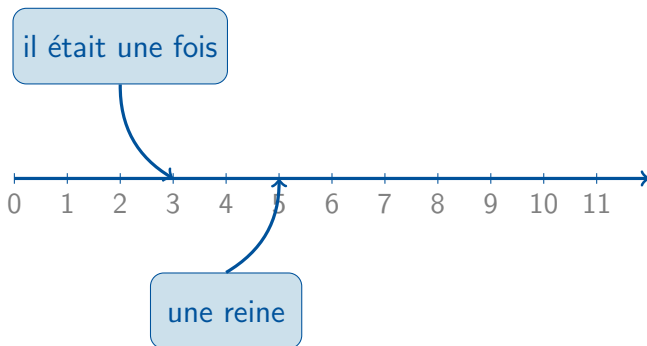
## Exemple 1: animations



Dans un beamer, utiliser les fonctions `\only` et `\uncover`.

```
\uncover<2->{ \node[event] (debut) at (4,-\z) {il etait une  
fois}; }
```

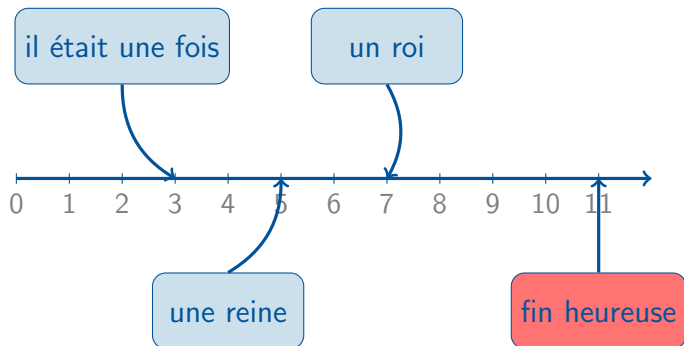
## Exemple 1: animations



Dans un beamer, utiliser les fonctions `\only` et `\uncover`.

```
\uncover<2->{ \node[event] (debut) at (4,-\z) {il etait une  
fois}; }  
\uncover<3->{ ... "une reine"... };
```

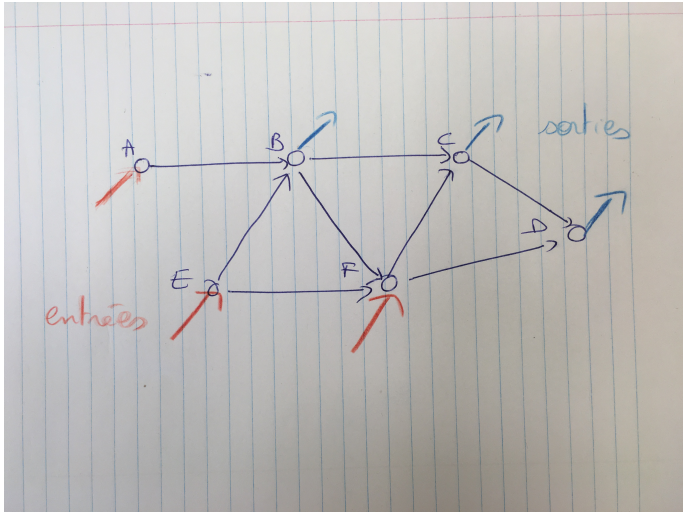
# Exemple 1: animations



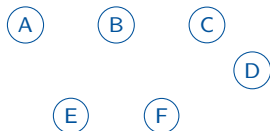
Dans un beamer, utiliser les fonctions `\only` et `\uncover`.

```
\uncover<2->{ \node[event] (debut) at (4,-\z) {il etait une
fois}; }
\uncover<3->{ ... "une reine"... };
```

# Exemple 2

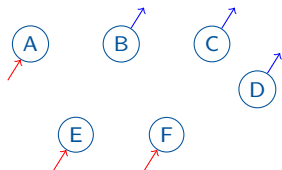


## Exemple 2



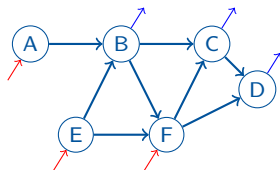
```
\def\hA{2}
\tikzstyle{noeud}=[draw,circle,inner sep=2pt]
\node[noeud] (A) at (0,\hA) {A};
\node[noeud] (B) at (2,\hA) {B};
\node[noeud] (C) at (4,\hA) {C};
\node[noeud] (D) at (5,\hA/2){D};
\node[noeud] (E) at (1,0){E};
\node[noeud] (F) at (3,0){F};
```

## Exemple 2



```
\foreach \n in {B,C,D}
    \draw[->,blue] (\n)---++(0.5,0.8);
\foreach \n in {A,E,F}
    \draw[<-,red] (\n)---++(-0.5,-0.8);
```

## Exemple 2



```

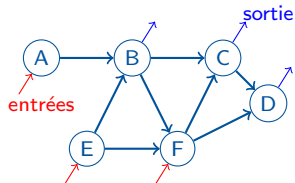
\foreach \n in {B,C,D}
    \draw[->,blue] (\n)---++(0.5,0.8);
\foreach \n in {A,E,F}
    \draw[<-;,red] (\n)---++(-0.5,-0.8);

\foreach \a/\b in {A/B,B/C,E/F,F/D,E/B,B/F,F/C,C/D}
    \draw[->,thick] (\a)--(\b);

```

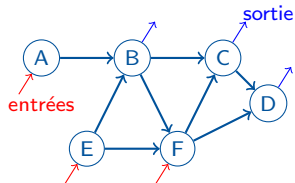


## Exemple 2



```
\node[red] at (0,\hA/2) {entrées};  
\node[blue] at (5,\hA+1) {sortie};
```

## Exemple 2



```
\node[red] at (0,\hA/2) {entrées};  
\node[blue] at (5,\hA+1) {sortie};
```

# Résumé

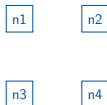
- définir des noeuds ou des coordonnées  
`\coordinate (A) at (0,1);` (utile si on veut seulement réutiliser le point)
- définir des styles `tikzstyle` (voir aussi `\tikzset`)
- définir des variables `def` ou `newcommand`
- coordonnées relatives `--++`
- afficher la grille `grid`
- utiliser les boucles: `\foreach`

## 5. La puissance de TikZ

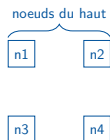
## 5. La puissance de TikZ

Varia

# Accolades

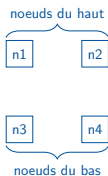


# Accolades



```
\usetikzlibrary{decorations.pathreplacing}
\draw[decorate, decoration={brace,amplitude=10pt, raise=2pt}] (n1.north
west) -- (n2.north east)
node [midway, below=13pt] {noeuds du haut};
```

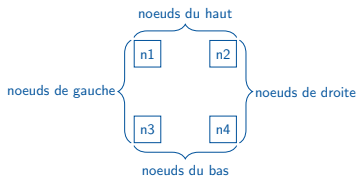
# Accolades



```
\usetikzlibrary{decorations.pathreplacing}  
\draw[decorate, decoration={brace,amplitude=10pt, raise=2pt, mirror }]  
(n3.south west) -- (n4.south east)  
node [midway, below=13pt] {noeuds du bas};
```



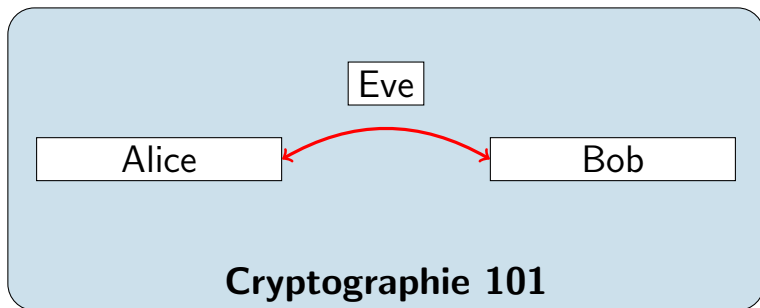
# Accolades



```
\usetikzlibrary{decorations.pathreplacing}
\draw[decorate, decoration={brace,amplitude=10pt, raise=2pt}] (n3.south
west) -- (n1.north west)
node [midway, left=10pt] {noeuds de gauche};

\draw[decorate, decoration={brace,amplitude=10pt,mirror, raise=2pt}]
(n4.south east) -- (n2.north east)
node [midway, right= 10pt] {noeuds de droite};
```

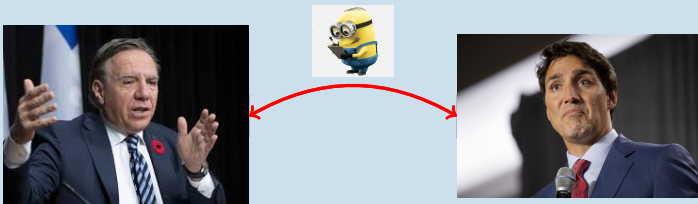
# Intégrer une figure jpg/png



# Intégrer une figure jpg/png

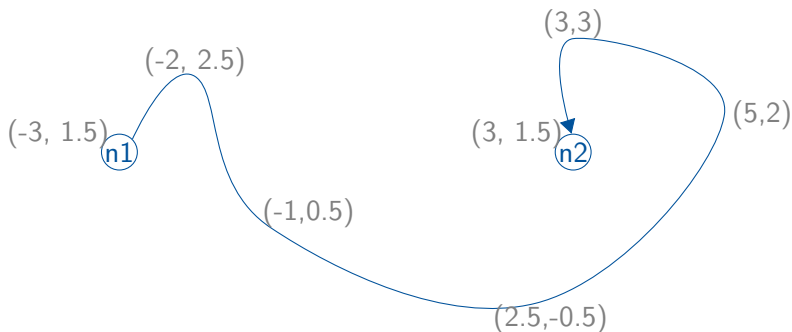
“Remplir” le noeud par une image:

```
\node[inner sep=0pt] (legault) at (0,0)  
{\includegraphics[width=2cm]{legault.jpg}};
```



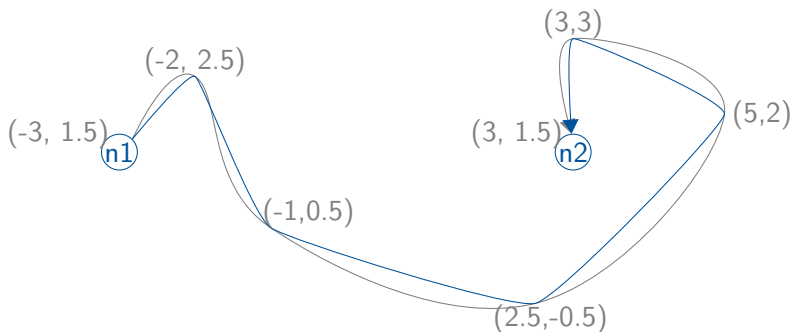
**Cryptographie 101**

# Arc de forme arbitraire



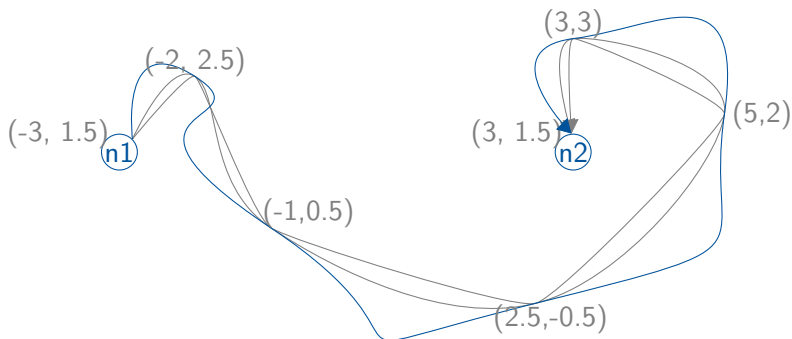
```
\draw[->, -triangle 60] plot[smooth, tension=0.7] coordinates {(n1.45)  
(-2,2.5) (-1,0.5) (2.5,-0.5) (5,2) (3,3) (n2.95)};
```

# Arc de forme arbitraire



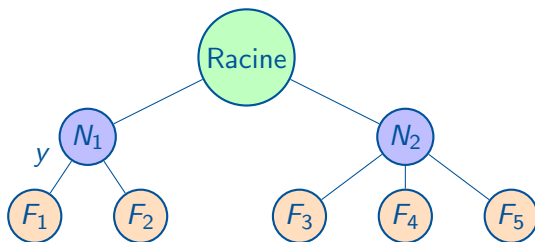
```
\draw[->, -triangle 60] plot[smooth, tension=0.2] coordinates {(n1.45)  
(-2,2.5) (-1,0.5) (2.5,-0.5) (5,2) (3,3) (n2.95)};
```

# Arc de forme arbitraire



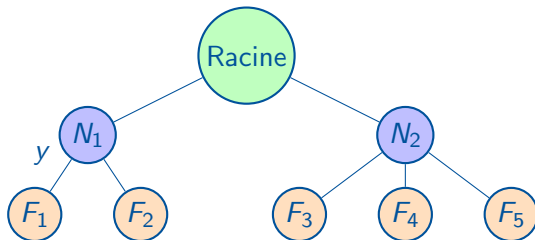
```
\draw[->, -triangle 60] plot[smooth, tension=2] coordinates {(n1.45)
(-2,2.5) (-1,0.5) (2.5,-0.5) (5,2) (3,3) (n2.95)};
```

# Construire un arbre



```
\node[noeud=green] {Racine}
  child { node[noeud=blue]{N1}
    child { node[noeud]{F1} }
    child { node[noeud]{F2} }
  }
  child { node[noeud=blue]{N2}
    child { node[noeud]{F3} }
    child { node[noeud]{F4} }
    child { node[noeud]{F5} }
  };
```

# Construire un arbre



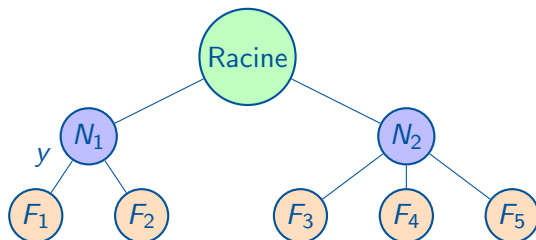
```

\node[noeud=green] {Racine}
  child { node[noeud=blue]{N1}
    child { node[noeud]{F1} edge from parent node[above left]
      {y}}
    child { node[noeud]{F2} }
  }
  child { node[noeud=blue]{N2}
    child { node[noeud]{F3} }
    child { node[noeud]{F4} }
    child { node[noeud]{F5} }
  }
};

```



# Construire un arbre



Gestion des niveaux:

```
\tikzstyle{level 1}=[level distance=1.5cm,sibling distance=6cm]
```

```
\tikzstyle{level 2}=[level distance=1.5cm,sibling distance=2cm]
```

## 5. La puissance de TikZ

Tracer des courbes et surfaces avec pgfplots

# pgfplots

## Description

- Package permettant de tracer des courbes.
- Axes automatiques.
- 2D/3D.

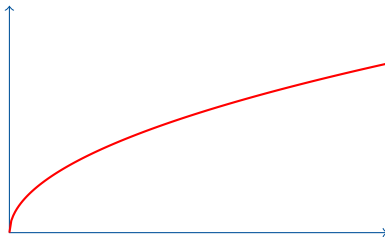
## Usage

```
\usepackage{pgfplots}  
\pgfplotsset{compat=newest}  
\usepackage{tikz-3dplot}
```

## Attention!

Il peut y avoir des conflits avec le package babel!

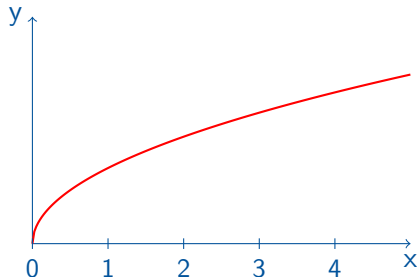
# Tracer une fonction (méthode rapide)



```
\draw [<->] (0,3) -- (0,0) -- (5,0);  
\draw[red, thick, domain=0:5] plot[samples=200] (\x, {sqrt(\x)});
```

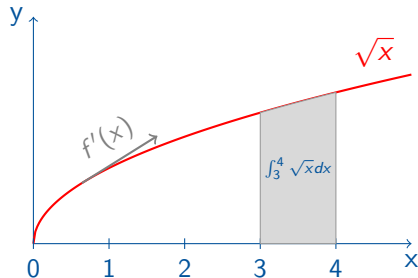
- samples= nb de points utilisés dans le tracé (défaut=25)
- Fonctions définis: sqrt, cos, sin, tan, exp, ln, floor ...
- Fonction paramétriques, polaires

# Tracer une fonction (méthode rapide)



```
\draw [<->] (0,3) -- (0,0) -- (5,0);  
\draw[red, thick, domain=0:5] plot[samples=200] (\x, {sqrt(\x)});  
  
\node[below] at (5,0) {x};  
\node[left] at (0,3) {y};  
\foreach \x in {0,1,...,4} \draw(\x,2pt)----+(0,-4pt)  
node[below]{\small\x};
```

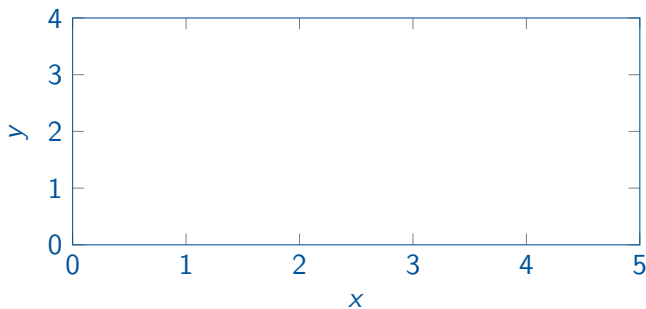
# Tracer une fonction (méthode rapide)



```
\draw [<->] (0,3) -- (0,0) -- (5,0);
\draw[red, thick, domain=0:5] plot[samples=200] (\x, {sqrt(\x)});

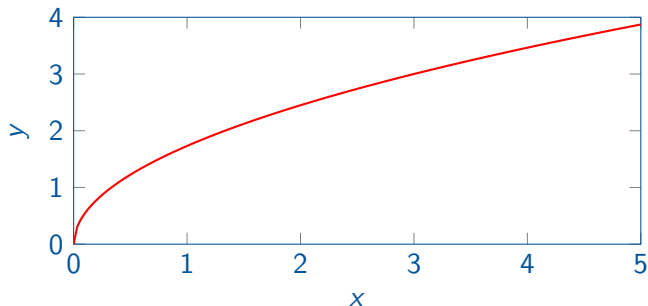
\filldraw[fill=gray!30,draw=gray!80] (3,0) -- (3,1.73) --
plot[domain=3:4] (\x, {sqrt(\x)}) -- (4,0)--cycle;
```

# Avec pgfplots



```
\begin{axis}[  
  x=1.5cm, y=0.75cm,  
  xtick distance = 0.5, ytick distance = 1,  
  xlabel = x, ylabel = y,  
  xmin = 0, xmax = 5, ymin = 0, ymax = 4,  
  samples = 160, domain = 0:5,  
]  
  Courbes à afficher...  
\end{axis}
```

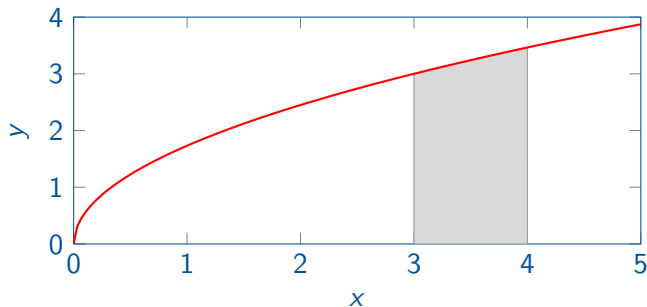
# Avec pgfplots



```
\addplot[red, thick, mark=none, ] {sqrt(3*x)};
```

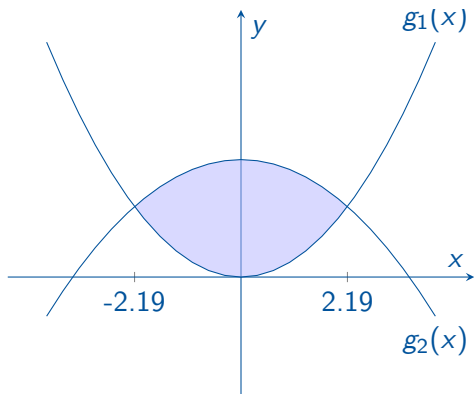


# Avec pgfplots



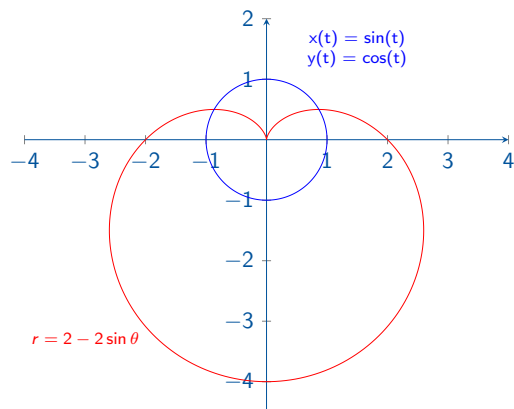
```
\filldraw[fill=gray!30,draw=gray!80] (3,0)--(3,1.73) --plot[domain=3:4]
(\x, {\sqrt{3*\x}}) --(4,0)--cycle;
\addplot[red, thick, mark=none, ] {\sqrt{3*x}};
```

# Aire entre deux courbes



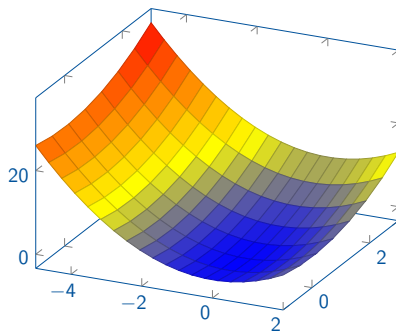
```
\addplot[name path=F,domain={-4:4}] {-(1/6)*x^2+2} node[pos=1,
below]{ $g_2(x)$ };
\addplot[name path=G,domain={-4:4}] {0.25*x^2}node[pos=1, above]{ $g_1(x)$ };
\addplot[blue!30, opacity=0.5]fill between[of=F and G, soft
clip={domain=-2.19:2.19}];
```

# Coordonnées polaires, équations paramétriques



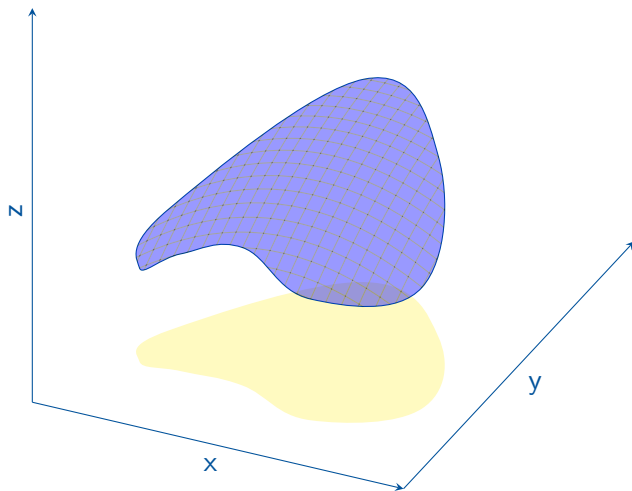
```
\addplot[domain=0:360, data cs=polar, ...] (x,{2 - 2*sin(x)});  
\addplot[domain=0:360, ...] ({sin(x)},{cos(x)});
```

# Surfaces 3D

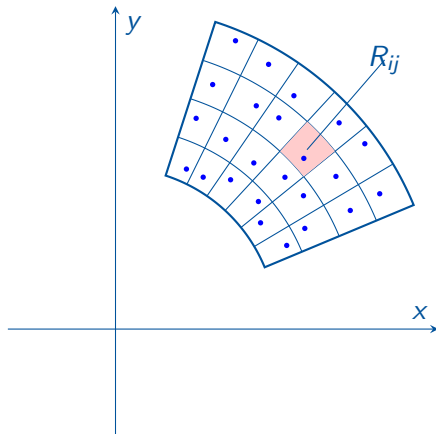


```
\addplot3 [domain=-5:2, domain y = -1:3, samples = 20, samples y = 8,  
surf] {x^2 + y^2};
```

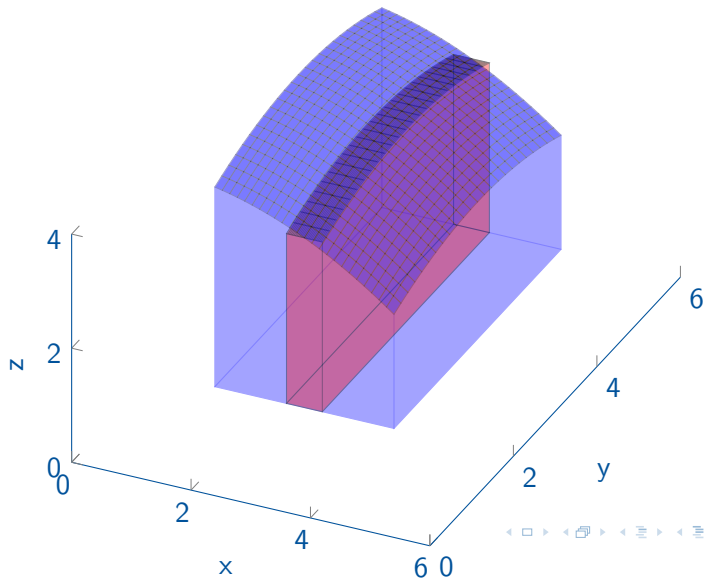
# Avec de la pratique...



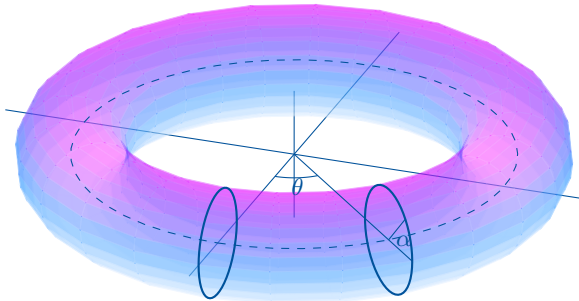
# Avec de la pratique...



# Avec de la pratique...

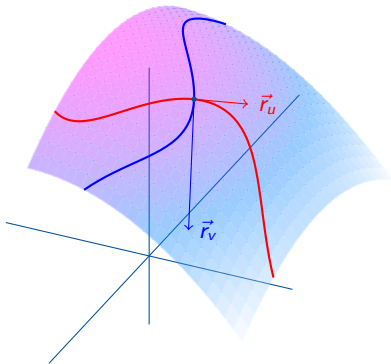


# Avec de la pratique...

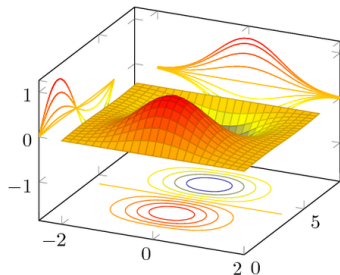
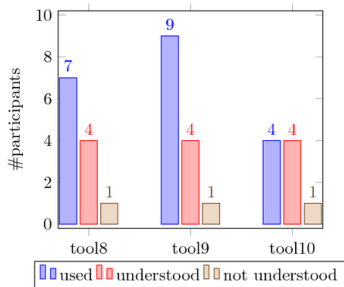




# Avec de la pratique...



# Ressources additionnelles



Exemples : <http://pgfplots.net/>

Tutoriel (fr) : <http://bertrandmasson.free.fr/index.php?article28/comment-faire-de-beaux-graphiques-avec-tikz-et-pgfplots>

## 5. La puissance de TikZ

Logiciels



# TikzEdt

## Avantages

## Inconvénients

# TikzEdt

## Avantages

- Éditeur WYSIWYG
- Permet la modification du code
- Simple d'utilisation

## Inconvénients

# TikZEdt

## Avantages

- Éditeur WYSIWYG
- Permet la modification du code
- Simple d'utilisation

## Inconvénients

- Windows seulement
- Ne supporte pas les animations Beamer
- Incorporation de bibliothèques difficiles

# TikZEdt

## Avantages

- Éditeur WYSIWYG
- Permet la modification du code
- Simple d'utilisation

## Verdict :

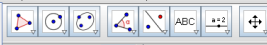
Placer les éléments avec TikZEdt.  
Styliser dans votre éditeur  $\text{\LaTeX}$ .

## Inconvénients

- Windows seulement
- Ne supporte pas les animations Beamer
- Incorporation de bibliothèques difficiles



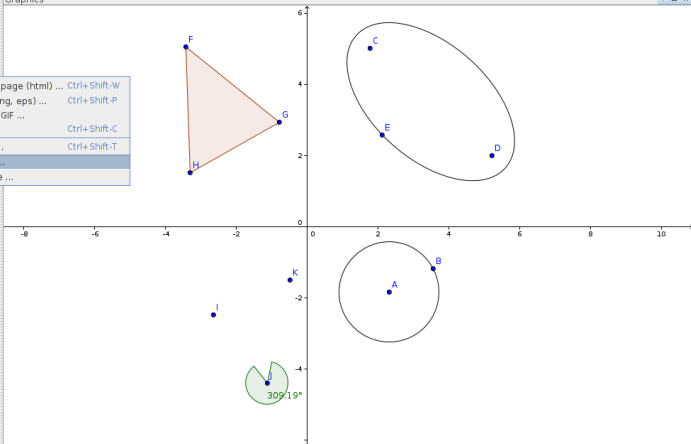
- New Window Ctrl-N
- New
- Open ... Ctrl-O
- Open Webpage ...
- Open Recent
- Save Ctrl-S
- Save As ...
- Share...
- Export
- Print Preview Ctrl-P
- Close Alt-F4



**Move**  
Drag or select objects (Esc)

- Dependent Objects
- c :  $(x - 2.32)^2 + (y +$
  - d :  $78.88x^2 + 83.11xy$
  - f = 2.89
  - g = 3.54
  - h = 3.39
  - poly1 = 4.55
  - $\alpha = 309.19^\circ$

- Dynamic Worksheet as Webpage (html) ... Ctrl+Shift-W
- Graphics View as Picture (png, eps) ... Ctrl+Shift-P
- Graphics View as Animated GIF ...
- Graphics View to Clipboard Ctrl+Shift-C
- Graphics View as PSTricks ... Ctrl+Shift-T
- Graphics View as PGF/TikZ ...
- Graphics View as Asymptote ...



# Geogebra

## Avantages

## Inconvénients

# Geogebra

## Avantages

- Éditeur WYSIWYG
- Maths
- Figure géométrique
- Facile de faire des figures complexes ou en 3d

## Inconvénients

# Geogebra

## Avantages

- Éditeur WYSIWYG
- Maths
- Figure géométrique
- Facile de faire des figures complexes ou en 3d

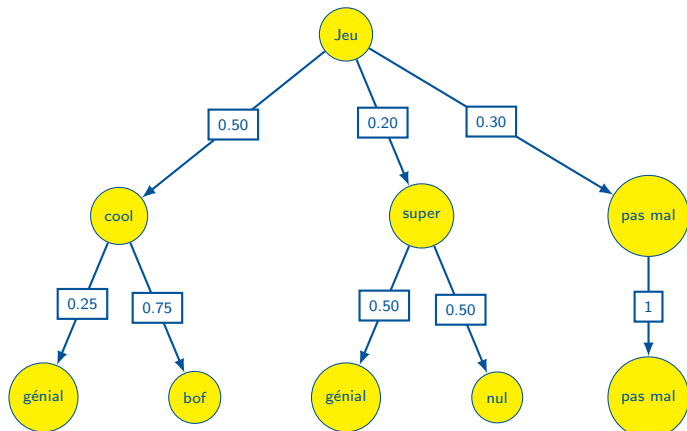
## Inconvénients

- Donne un code difficile à modifier
- Contrôle de certains attributs long

# Construire un arbre très rapidement

Outil de création rapide d'un arbre en ligne:

<http://math.et.info.free.fr/TikZ/Arbres.html>



# Références

- Wikibooks sur TikZ:  
<https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ>
- CTAN sur TikZ incluant le pdf du livre officiel de Crémer et autre documentation: <https://www.ctan.org/pkg/pgf>
- base de données TeXample TikZ remplie d'exemples et d'inspirations :  
<https://texample.net/tikz/>



Jacques Crémer.

*A very minimal introduction to TikZ.*

2011.



Till Tantau.

*The TikZ and PGF Packages.*

Institut für Theoretische Informatik, November 2012.



Gérard Tisseau and Jacques Duma.

*TikZ pour l' impatient.*

2012.

# Merci pour votre attention!

©Luciano Costa, Lucie Desfontaines,  
Frédéric Quesnel, GERAD