Tikz introduction: how to draw a platypus

Emma Caizergues[1,2]

[1] Nokia Bell Labs, Massy, France
[2] Université Paris-Dauphine, Paris, France

What's Tikz ?

Informal definition

- A language to make vectorial graphics

What's Tikz ?

Informal definition

- A language to make vectorial graphics
- You can type it directly in your .tex file

What's Tikz ?

Informal definition

- A language to make vectorial graphics
- You can type it directly in your .tex file

Main advantages

- Adaptability

What's Tikz ?

Informal definition

- A language to make vectorial graphics
- You can type it directly in your .tex file

Main advantages

- Adaptability
- Practical with slides

What's Tikz ?

Informal definition

- A language to make vectorial graphics
- You can type it directly in your .tex file

Main advantages

- Adaptability
- Practical with slides
- Compatible with your chosen notations

What's Tikz ?

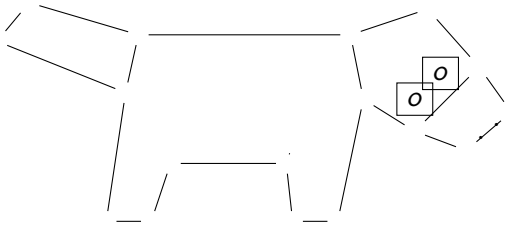
Informal definition

- A language to make vectorial graphics
- You can type it directly in your .tex file

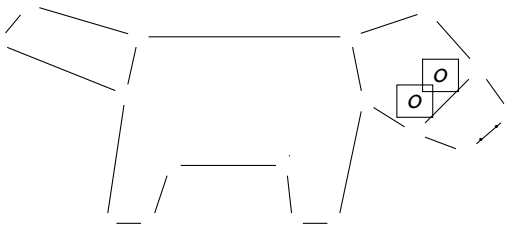
Main advantages

- Adaptability
- Practical with slides
- Compatible with your chosen notations
- A lot of literature

Overview

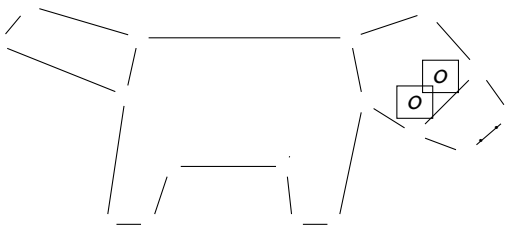


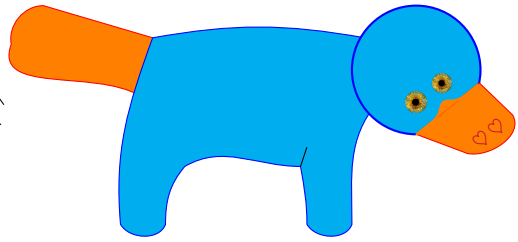
Overview



- First steps with graphics in general

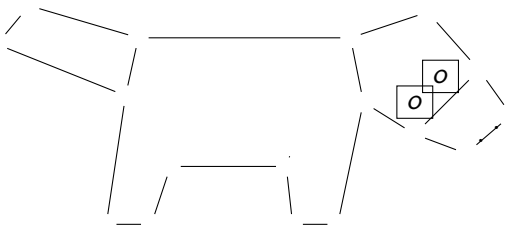
Overview

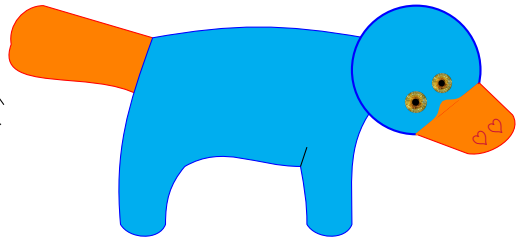




- First steps with graphics in general

Overview





- First steps with graphics in general

- Additional functionalities
- "Advanced features"
- Plotting functions

Import package and create figure

```
\usepackage{tikz}
\usepackage{pgfplots}
```

Import package and create figure

```
\usepackage{tikz}
\usepackage{pgfplots}

\usetikzlibrary{positioning}
\usetikzlibrary{automata,tree}
```

Import package and create figure

```
\usepackage{tikz}
\usepackage{pgfplots}

\usetikzlibrary{positioning}
\usetikzlibrary{automata,tree}

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

First steps: nodes

```
\node (a) at (0,0) {} ;
```

First steps: nodes

```
\node (a) at (0,0) {} ;

\node (a) at (0,0) {$a$} ;
```

First steps: nodes

```
\node (a) at (0,0) {} ;

\node (a) at (0,0) {$a$} ;

\node[draw] (a) at (0,0) {$a$} ;
```

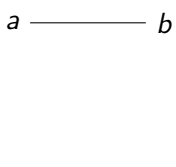
First steps: nodes


```
\node (a) at (0,0) {} ;

\node (a) at (0,0) {$a$} ;

\node[draw] (a) at (0,0) {$a$} ;

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

First steps: nodes


```
\node (a) at (0,0) {} ;

\node (a) at (0,0) {$a$} ;

\node[draw] (a) at (0,0) {$a$} ;

\node[circle,draw] (a) at (0,0) {$a$} ;

\node[fill=yellow,text=black,circle,draw=red,thick
    ] (a) at (0,0) {$a$} ;
```

First steps: connecting nodes




```
\node (a) at (0,1)  {$a$} ;
\node (b) at (2,1)  {$b$} ;
\node (c) at (1,-1) {$c$} ;

\draw (a) -- (b) ;
```

First steps: connecting nodes


```
\node (a) at (0,1)  {$a$} ;
\node (b) at (2,1)  {$b$} ;
\node (c) at (1,-1) {$c$} ;

\draw[->] (a) -- (b) ;
```

c

First steps: connecting nodes


```
\node (a) at (0,1)  {$a$} ;
\node (b) at (2,1)  {$b$} ;
\node (c) at (1,-1) {$c$} ;

\draw[thick] (a)--(b) node[pos=0.3, above]{$1$} ;
```

First steps: connecting nodes


```
\node (a) at (0,1)  {$a$} ;
\node (b) at (2,1)  {$b$} ;
\node (c) at (1,-1) {$c$} ;

\draw (a) -- (a) ;
```

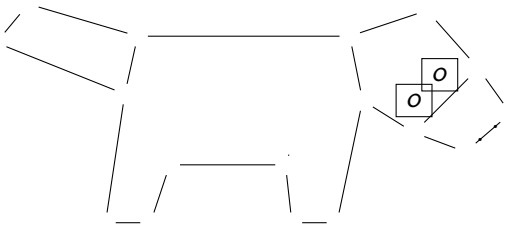
First steps: connecting nodes


```
\node (a) at (0,1)  {$a$} ;
\node (b) at (2,1)  {$b$} ;
\node (c) at (1,-1) {$c$} ;

\draw (a) edge[loop] (a) ;
```

First steps: connecting nodes


```
\node (a) at (0,1)  {$a$} ;
\node (b) at (2,1)  {$b$} ;
\node (c) at (1,-1) {$c$} ;

\draw (a) edge[out=-45,in=45,loop] (a) ;
```

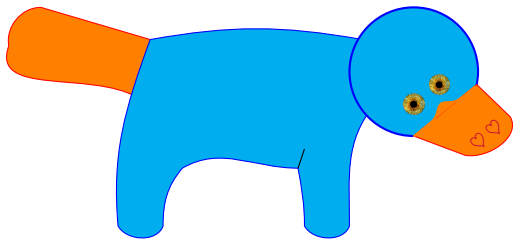
My platypus



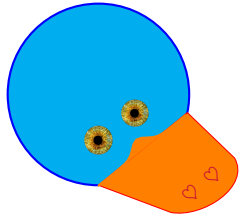
```
\node (a) at (-0.8,1.5) {};
\node (b) at (-1.1,0.6) {};
\node (c) at (-1.8,0.6) {};
\node (d) at (-1.5,2.6) {};
\node (e) at (-1.3,3.5) {};
\node (f) at (2,3.5)    {};
\node (g) at (2.2,2.5)  {};
\node (h) at (1.8,0.6)  {};
\node (i) at (1.1,0.6)  {};
\node (j) at (1,1.5)    {};
\node (k) at (1.1,1.8)  {};
```

```
\draw (a) -- (b) -- (c) -- (d) -- (e) --
(f) -- (g) -- (h) -- (i) -- (j) --(k) ;
```

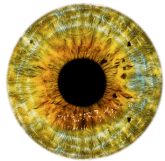
My improved platypus



My improved platypus



Additional features: images, fancy text, rotation, shift



```
\node (a) at (0,0) {\includegraphics[scale=0.2]{
    oeil.png}} ;
```

Additional features: images, fancy text, rotation, shift

```
\node (a) at (0,0) {$\heartsuit$} ;
```

Additional features: images, fancy text, rotation, shift

```
\node[rotate=45] (a) at (0,0) {$\heartsuit$} ;
```

Additional features: images, fancy text, rotation, shift

```
\node[rotate=45,xshift=1cm,yshift=2cm] (a) at
    (0,0) {$\heartsuit$} ;
```

Additional features: images, fancy text, rotation, shift

```
\node[xshift=1cm,yshift=2cm,rotate=45] (a) at
    (0,0) {$\heartsuit$} ;
```

Additional features: circles, squares, curvated edges

```
\draw [color=blue,fill=cyan,thick] (3+7.8,2) arc [
    radius=1, start angle=270, end angle= -30];
```

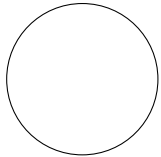
Additional features: circles, squares, curvated edges

```
\draw (0,0) arc [radius=0.7, start angle=0, end
    angle= -180];
```

Additional features: circles, squares, curvated edges

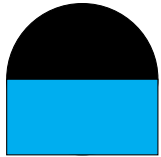
```
\draw [color=blue,fill=cyan,thick] (0,0) arc [
    radius=1, start angle=270, end angle= -30];
```

Additional features: circles, squares, curvated edges



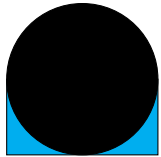
```
\draw (0,0) circle (1)  ;
```

Additional features: circles, squares, curvated edges



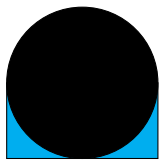
```
\draw[fill=black] (0,0) circle (1)  ;
\draw[fill=cyan] (-1,-1) rectangle (1,1) ;
```

Additional features: circles, squares, curvated edges



```
\draw[fill=cyan] (-1,-1) rectangle (1,1) ;
\draw[fill=black] (0,0) circle (1)  ;
```

Additional features: circles, squares, curvated edges



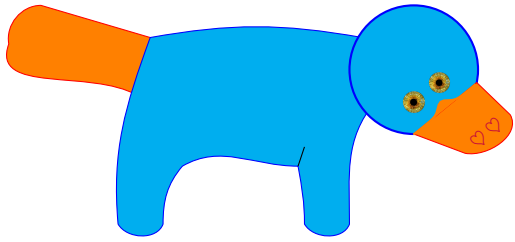
```
\def\mycutefigure{
        \draw[fill=cyan] (-2,-1) rectangle (0,0) ;
        \draw[fill=black] (-1,0) circle (1) ;
        }

\begin{scope}[yshift=1cm]
        \mycutefigure
\end{scope}
```

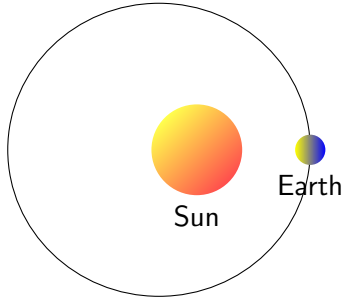
Additional features: circles, squares, curvated edges

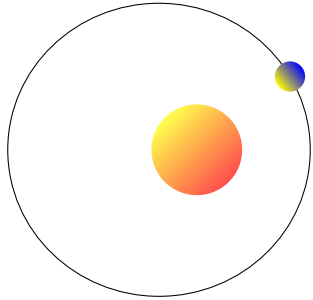
Additional features: circles, squares, curvated edges

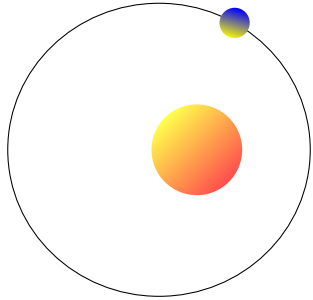
```
\draw[color=red,fill = orange] (0.8,1.7) to[out
    =350,in=300] (1.5,2.3) ;
```

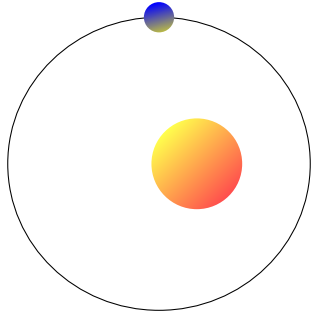
My improved platypus

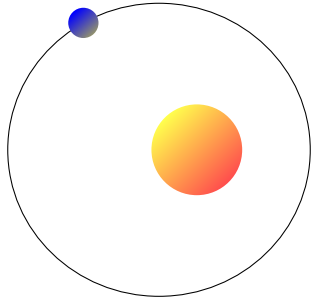


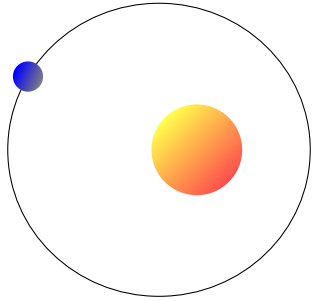

Earth
Sun

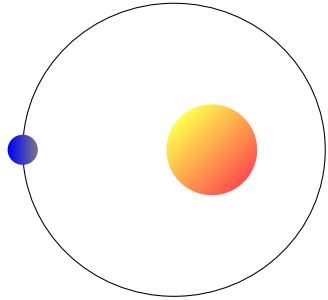


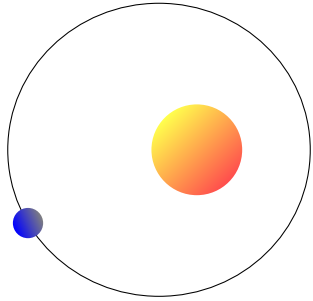


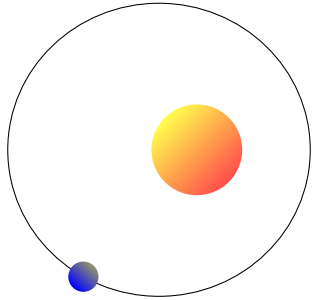


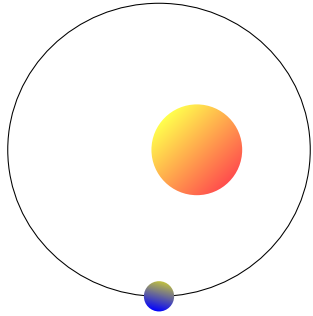


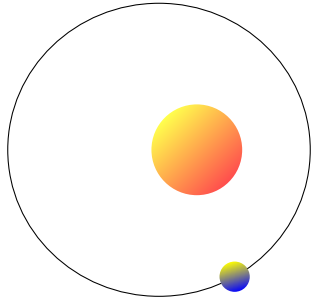


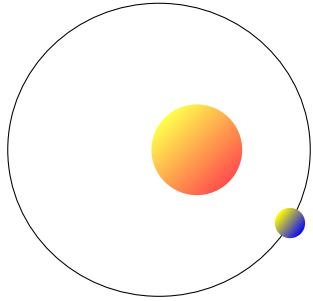


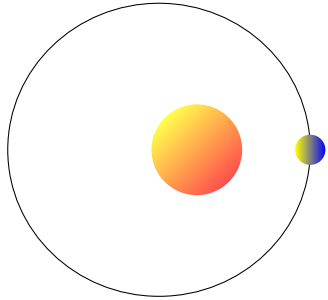


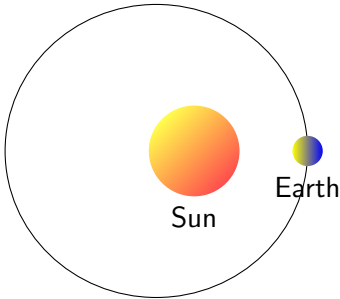




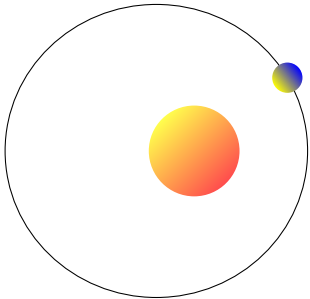




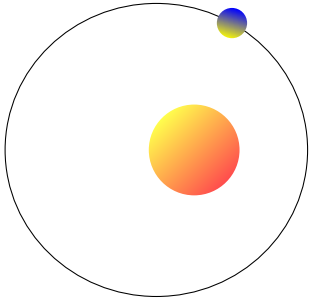




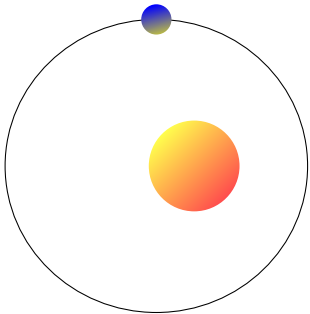

```
\setbeamercovered{invisible}
\pgfmathsetmacro{\Sunradius}{0.3}    % Sun
    radius
\pgfmathsetmacro{\Earthradius}{0.1} %
    Earth radius
\pgfmathsetmacro{\e}{0.25}           %
    Excentricity of the elliptical orbit
\pgfmathsetmacro{\b}{sqrt(1-\e*\e)} %
    Minor radius (major radius = 1)
```



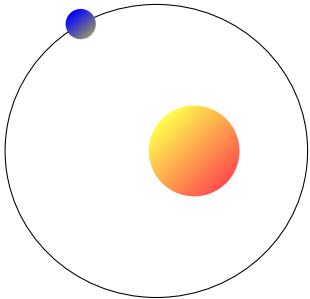
```
% Draw the Sun at the right-hand-side
    focus
\shade[
top color=yellow!70,
bottom color=red!70,
shading angle={45},
] ({sqrt(1-\b*\b)},0) circle (\Sunradius);
\visible<1>{
        \draw ({sqrt(1-\b*\b)},-\Sunradius
            ) node[below] {Sun};
}
```



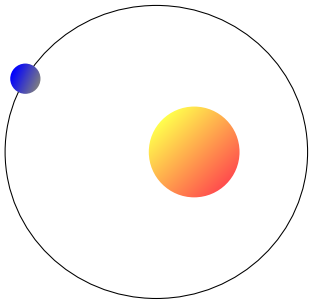
```
% Draw the elliptical path of the Earth.
\draw[thin] (0,0) ellipse (1 and {\b});
```



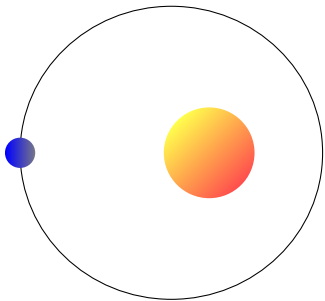
```
% This function computes the direction in
    which light hits the Earth.
\pgfmathdeclarefunction{f}{1}{%
\pgfmathparse{
        ((-\e+cos(#1))<0) * ( 180 + atan(
            \b*sin(#1)/(-\e+cos(#1)) ) )
                +
        ((-\e+cos(#1))>=0) * ( atan( \b*
            sin(#1)/(-\e+cos(#1)) ) )
        }
}
```



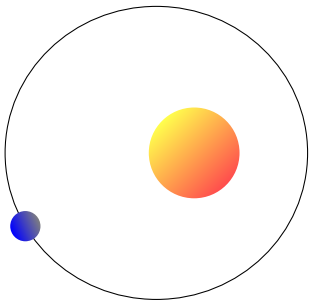
```
% This function computes the distance
    between Earth and the Sun, which is
    used to calculate the varying
    radiation intensity on Earth.
\pgfmathdeclarefunction{d}{1}{%
\pgfmathparse{ sqrt((-\e+cos(#1))*(-\e+cos
    (#1))+\b*sin(#1)*\b*sin(#1)) }
}
```



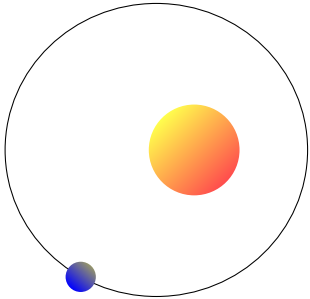
```
% This function computes the distance
    between Earth and the Sun,
% which is used to calculate the varying
    radiation intensity on Earth.
\pgfmathdeclarefunction{d}{1}{%
\pgfmathparse{ sqrt((-\e+cos(#1))*(-\e+cos
    (#1))+\b*sin(#1)*\b*sin(#1)) }
}
```

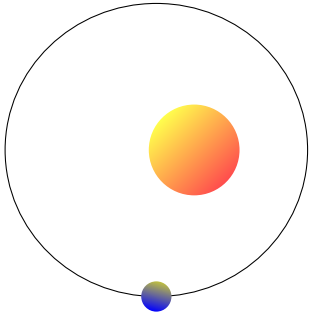


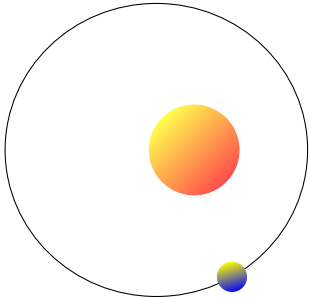
```
% Produces a series of frames showing one
    revolution
\pgfmathtruncatemacro{\N}{12}
\foreach \k in {0,1,...,\N}{
\pgfmathsetmacro{\theta}{360*\k/\N}
\pgfmathsetmacro{\radiation}{100*(1-\e)/(d
    (\theta)*d(\theta))}
\colorlet{Earthlight}{yellow!\radiation!
    blue}
\pgfmathparse{int(\k+1)}
\onslide<\pgfmathresult>{
\shade[top color=Earthlight, bottom color=
    blue, shading angle={90+f(\theta)}] ({
    cos(\theta)},{\b*sin(\theta)}) circle
    (\Earthradius);}}
```

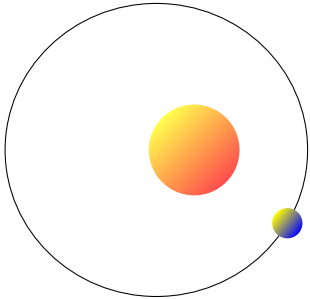


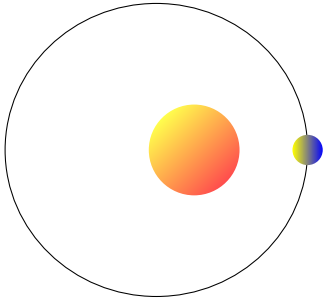
```
% Produces a series of frames showing one
    revolution
\pgfmathtruncatemacro{\N}{12}
\foreach \k in {0,1,...,\N}{
\pgfmathsetmacro{\theta}{360*\k/\N}
\pgfmathsetmacro{\radiation}{100*(1-\e)/(d
    (\theta)*d(\theta))}
\colorlet{Earthlight}{yellow!\radiation!
    blue}
\pgfmathparse{int(\k+1)}
\onslide<\pgfmathresult>{
\shade[top color=Earthlight, bottom color=
    blue, shading angle={90+f(\theta)}] ({
    cos(\theta)},{\b*sin(\theta)}) circle
    (\Earthradius);
\visible<1>{
\draw ({cos(\theta)},{\b*sin(\theta)-\
    Earthradius}) node[below] {Earth};}}
```



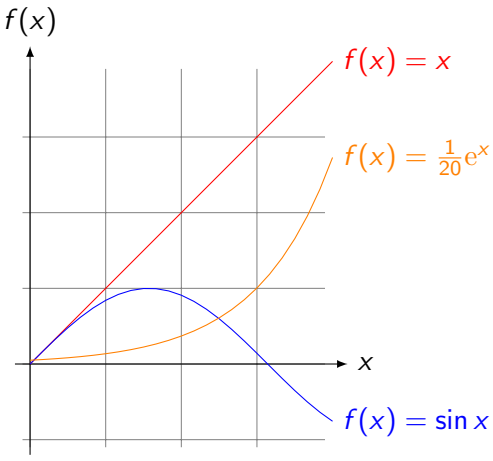






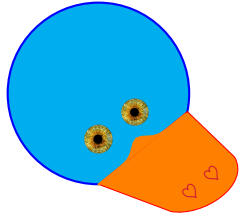


Plotting functions




```
\begin{tikzpicture}[domain=0:4]
        \draw[very thin,color=gray]
            (-0.1,-1.1) grid (3.9,3.9);

        \draw[->] (-0.2,0) -- (4.2,0) node
            [right] {$x$};
        \draw[->] (0,-1.2) -- (0,4.2) node
            [above] {$f(x)$};

        \draw[color=red] plot (\x,\x) node
            [right] {$f(x) =x$};

        \draw[color=blue] plot (\x,{sin(\x
             r)}) node[right] {$f(x) = \
            sin x$};

        \draw[color=orange] plot (\x
            ,{0.05*exp(\x)}) node[right]
            {$f(x) = \frac{1}{20} \mathrm
            e^x$};
\end{tikzpicture}
```

Plotting functions



Plotting functions

```
\draw[
rotate=39,scale=0.13,
orange,fill=orange,thick]
plot (\x, {exp((-5/6)*\x*\x)});
```

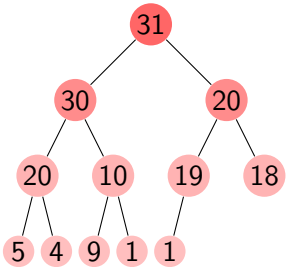
Plotting functions

```
\draw[xshift=11.3cm,yshift=2.4cm,
rotate=39,scale=0.13,
orange,fill=orange,thick]
plot (\x, {exp((-5/6)*\x*\x)});
```

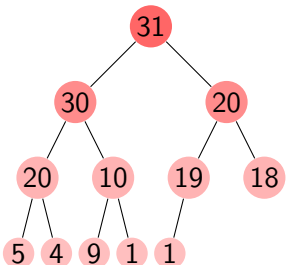
Some useful packages

- automata
- trees
- patterns
- petrinet

Trees

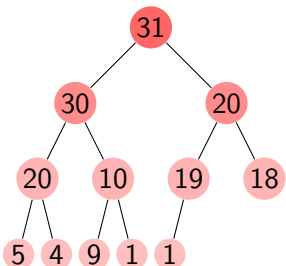


Trees



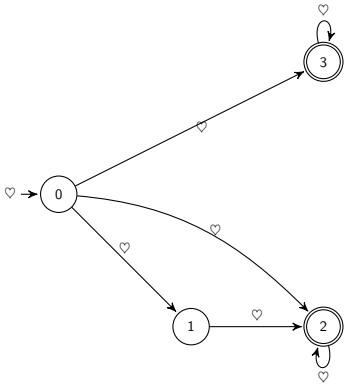

```
[level distance=10mm,
every node/.style={fill=red!60, circle, inner sep=1pt},
level 1/.style={sibling distance=20mm,nodes={fill=red!45}},
level 2/.style={sibling distance=10mm,nodes={fill=red!30}},
level 3/.style={sibling distance=5mm,nodes={fill=red!25}}]
```

Trees




```
[level distance=10mm,
every node/.style={fill=red!60, circle, inner sep=1pt},
level 1/.style={sibling distance=20mm,nodes={fill=red!45}},
level 2/.style={sibling distance=10mm,nodes={fill=red!30}},
level 3/.style={sibling distance=5mm,nodes={fill=red!25}}]

\node {31}
child {node {30} child {node {20} child {node {5}} child {
    node {4}}} child {node {10} child {node {9}} child {
    node {1}}}}
child {node {20} child {node {19} child {node {1}} child[
    missing]} child {node {18}}};
```

Automata




```
\begin{tikzpicture}[>=stealth',shorten >=1pt,auto,
    node distance=3.5 cm, scale = 0.6, transform
    shape]

\node[initial,initial text={start},state] (A)
    {$0$};
\node[state] (B) [right of=A,below of=A] {$1$};
\node[state,accepting] (D) [right of=B] {$2$};
\node[state,accepting] (C) [above of=D,above of=D]
     {$3$};

\path[->] (A) edge[below,right] node[align=center
    ]{$$} (C)
(A) edge[above] node[align=center]{$$} (B)
(A) edge[above,right,out=-5,in=135] node[align=
    center]{$$} (D)
(B) edge[above]       node[align=center]{$$} (D)
(C) edge[loop above] node[align=center]{$$} (C)
(D) edge[loop below] node[align=center]{$$} (D);

\end{tikzpicture}
```

Some resources

- Pgf/Tikz manual online: https://tikz.dev/
- LaTeX Wikibook: https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ
- Tutorials on Youtube
- Stack Overflow

Thank you :)

Questions ?

don't hesitate to contact me: emma.caizergues@dauphine.eu