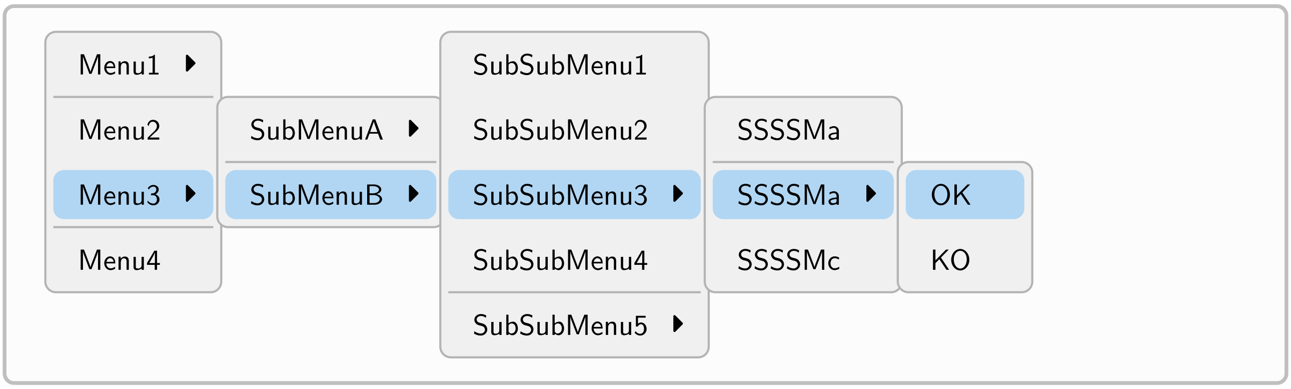
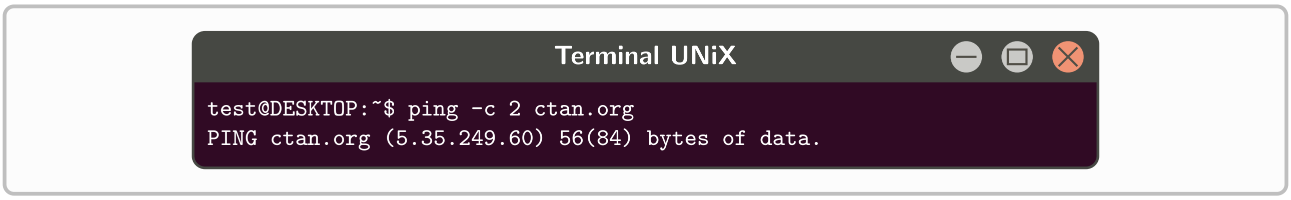
sim-os-menus

Simulate 'windows', 'terminal' or 'context menu' like in an OS.

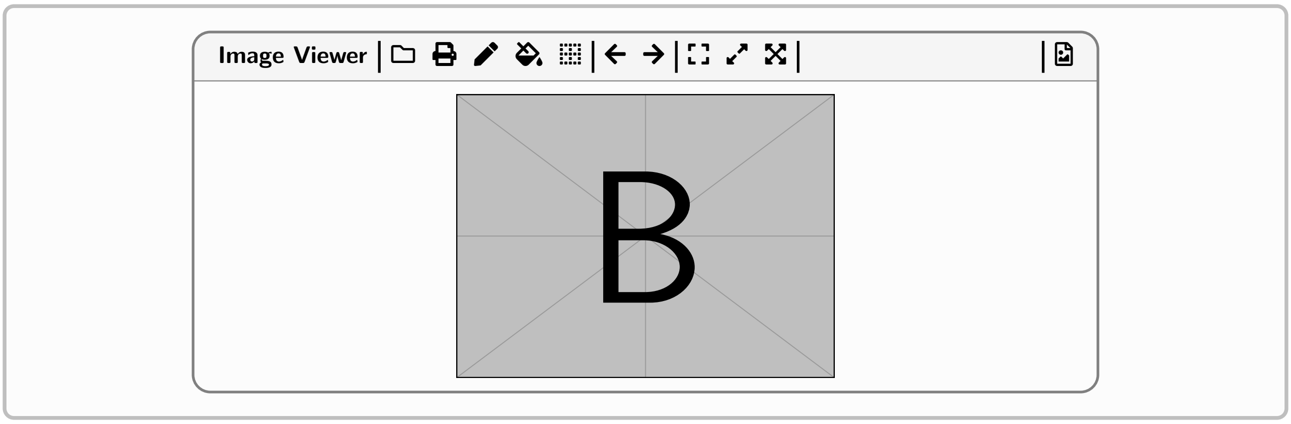
Version 0.1.6 -- 05/05/2025

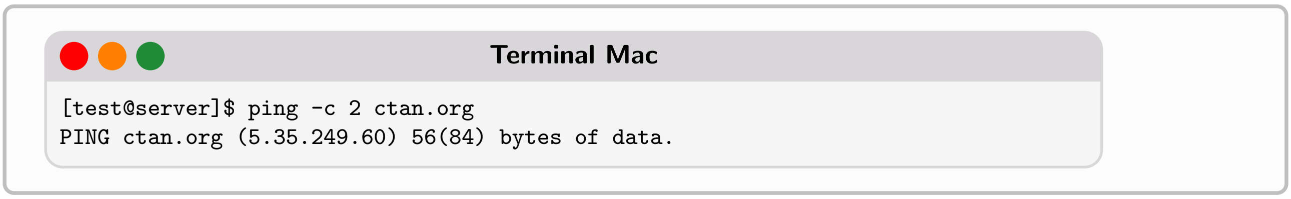
Cédric Pierquet
c pierquet -- at -- outlook . fr
https://forge.apps.education.fr/pierquetcedric/packages-latex





```
test@DESKTOP:~$ ping -c 2 ctan.org
PING ctan.org (5.35.249.60) 56(84) bytes of data.
```





```
[test@server]$ ping -c 2 ctan.org
PING ctan.org (5.35.249.60) 56(84) bytes of data.
```

Contents

1 Introduction

1.1 Description

With this package you can create context menu, or terminal, or doc viewer, like in an OS.
Global styles are mostly fixed, but some customizations are possible.

1.2 Loading

To load the package, simply use:

```
\usepackage{sim-os-menus}
```

The package loads the packages:

- `tikz` (with `calc,positioning`), `pgf`, `pgffor`;
- `calc`, `fontawesome5`;
- `simplekv`, `xintexpr`, `listofitems`, `xstring`;
- `settobox`, `tabularray`;
- `tcolorbox` (with `breakable,fitting,skins,listings,listingsutf8,hooks`).

For `fontawesome5` and `fontawesome6` :

```
\usepackage{sim-os-menus}                 %fa5, default
\usepackage[fa6]{sim-os-menus}            %with fa6
```

1.3 History

```
0.1.6: Bugfix + pre-compatibity with fa5/fa6
0.1.4: Mastodon or BlueSky posts 'like'
0.1.3: Fofders/Files like in explorer
0.1.2: Script editor viewer 'like'
0.1.1: French version of the commands
0.1.0: Initial version
```

2 The macros

2.1 Context menu

I order to create a context menu, the command is:

```
%----contextual menu
\ContextMenu[keys]{list of items}<tikz options>
```

Optional keys, between `[...]` are:

- `ColBack`: background color;
- `ColHL`: = highlight color;
- `Rounded`: boolean for rounded corners (`true` by default);
- `Font`: font for the items (`\normalsize\normalfont` by default);
- `ColItems`: color(s) for the items (`black` by default);
- `MarginV`: vertical margin of the lines (`6pt` by default);
- `MarginH`: horizontal margin of the lines (`12pt` by default);
- `Arrow`: character for the arrow (`\faCaretRight` by default);
- `ListSeps`: list for the possible sep lines (empty or for all the levels !);
- `ListIcons`: list for the possible icons (empty or for all the levels/items !);
- `ListOffsets`: list for the possible vertical offset of levels (from 2, ...!) (empty or for all the sub-levels !);
- `Icons`: boolean for icons (`false` by default);
- `Bar`: boolean for small vertical bar with icons (`true` by default);
- `Space`: horizontal space between levels (`-0.125` by default).

The mandatory argument, between `{...}`, is given as:

`item1A,item1B,... § item2A,itemp2B,... § ...`

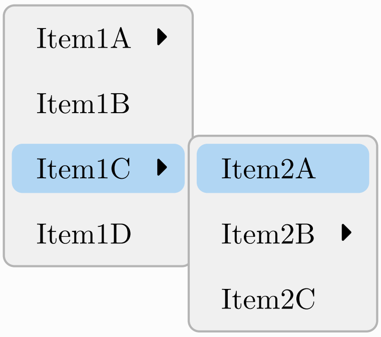
- if an item ends with `(*)`, this is the beginning of the next level (only one by level !);
- if an item ends with `(>)` (before optional `(*)`), an arrow is written at the end of the line.

A correct usage of the syntax is necessary for the code !

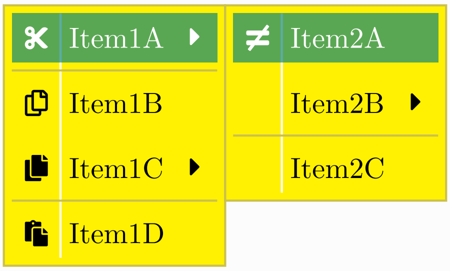
A few tips, due to `ListIcons`, `ListOffsets` and `ListSeps` keys, which are *sensitive*:

- `ListIcons` must have the same number of elements than the number of levels/items (with possible empty items);
- `ListSeps` must have the same number of elements than the number of levels (with possible empty items);
- `ListOffsets` must have the same number of elements than the numbers of sub-levels (with 0 si no offset !).

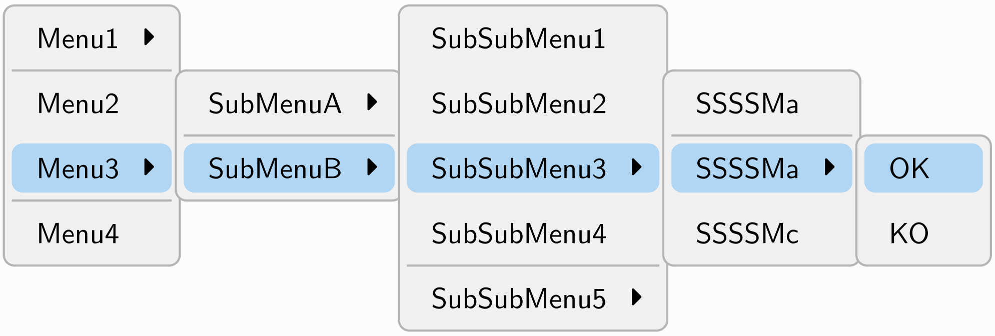
```
%default style
\ContextMenu{Item1A(>),Item1B,Item1C(>)(*),Item1D § Item2A(*),Item2B(>),Item2C}
```



```
%custom style
\ContextMenu[Rounded=false,ColBack=yellow,ColHL=teal,%
  ListSeps={1,3/2},ColItems={black/white},Icons,Space=0,%
  ListIcons={\faCut,\faIcon[regular]{copy},\faCopy,\faPaste / \faNotEqual}]
  {Item1A(>)(*),Item1B,Item1C(>),Item1D § Item2A(*),Item2B(>),Item2C}
```



```
\ContextMenu[Font=\sffamily,ListSeps={1,3/1/4/1/},ListOffsets={1,2,1,0}]{%
  Menu1(>),Menu2,Menu3(>)(*),Menu4 §
  SubMenuA(>),SubMenuB(>)(*) §
  SubSubMenu1,SubSubMenu2,SubSubMenu3(>)(*),SubSubMenu4,SubSubMenu5(>) §
  SSSSMa,SSSSMa(>)(*),SSSSMc §
  OK(*),KO
}
```



2.2 Terminal

In order to create a terminal (Win/UNiX/Mac), environments are:

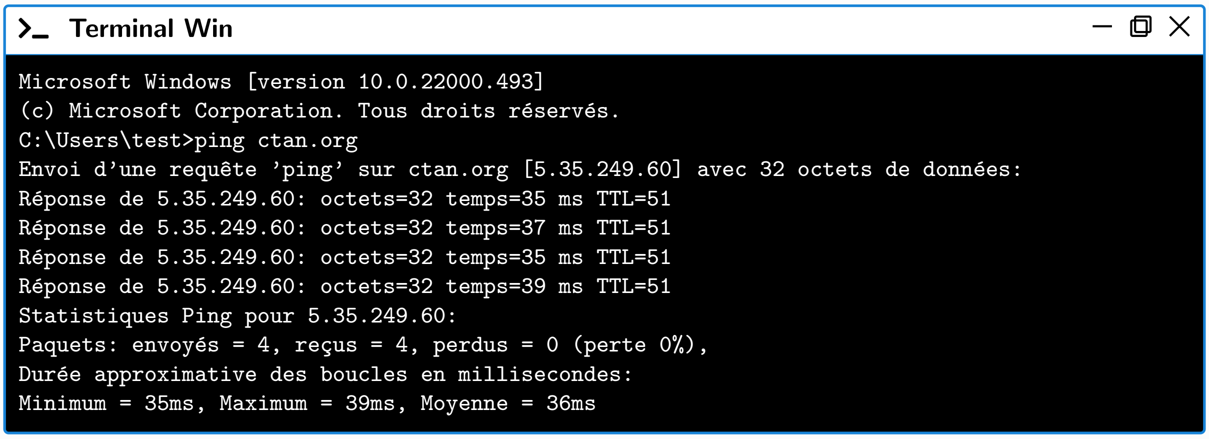
```
%----Windows like terminal
\begin{TermWin}[keys]{tcbox options}
    ...
\end{TermWin}

%----UNiX like terminal
\begin{TermUnix}[keys]{tcbox options}
    ...
\end{TermUnix}

%----OSX like terminal
\begin{TermMac}[keys]{tcbox options}
    ...
\end{TermMac}
```

Optional keys, between `[...]` are:

- `Title`: title of the terminal (`Terminal Win/UNiX/Mac` by default);
- `Align`: horizontal alignment of the box (`center` by default);
- `Width`: width of the box (`\linewidth` by default).

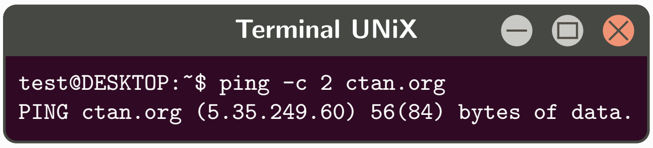
The mandatory argument, between `{...}`, are options to give to the tcbox.

```
\begin{TermWin}{}
Microsoft Windows [version 10.0.22000.493]
(c) Microsoft Corporation. Tous droits réservés.
C:\Users\test>ping ctan.org
Envoi d'une requête 'ping' sur ctan.org [5.35.249.60] avec 32 octets de données:
Réponse de 5.35.249.60: octets=32 temps=35 ms TTL=51
Réponse de 5.35.249.60: octets=32 temps=37 ms TTL=51
Réponse de 5.35.249.60: octets=32 temps=35 ms TTL=51
Réponse de 5.35.249.60: octets=32 temps=39 ms TTL=51
Statistiques Ping pour 5.35.249.60:
Paquets: envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes:
Minimum = 35ms, Maximum = 39ms, Moyenne = 36ms
\end{TermWin}
```



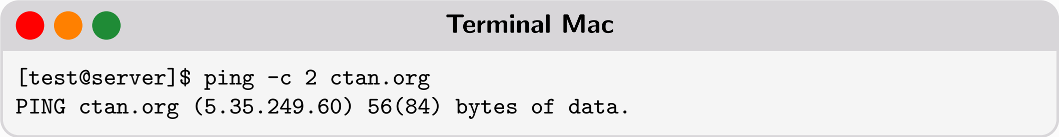
```
Microsoft Windows [version 10.0.22000.493]
(c) Microsoft Corporation. Tous droits réservés.
C:\Users\test>ping ctan.org
Envoi d'une requête 'ping' sur ctan.org [5.35.249.60] avec 32 octets de données:
Réponse de 5.35.249.60: octets=32 temps=35 ms TTL=51
Réponse de 5.35.249.60: octets=32 temps=37 ms TTL=51
Réponse de 5.35.249.60: octets=32 temps=35 ms TTL=51
Réponse de 5.35.249.60: octets=32 temps=39 ms TTL=51
Statistiques Ping pour 5.35.249.60:
Paquets: envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes:
Minimum = 35ms, Maximum = 39ms, Moyenne = 36ms
```

```
\begin{TermUnix}[Align=flush right]{hbox}
test@DESKTOP:~$ ping -c 2 ctan.org
PING ctan.org (5.35.249.60) 56(84) bytes of data.
\end{TermUnix}
```



```
test@DESKTOP:~$ ping -c 2 ctan.org
PING ctan.org (5.35.249.60) 56(84) bytes of data.
```

```
\begin{TermMac}[Width=14cm,Align=flush left]{}
[test@server]$ ping -c 2 ctan.org
PING ctan.org (5.35.249.60) 56(84) bytes of data.
\end{TermMac}
```



```
[test@server]$ ping -c 2 ctan.org
PING ctan.org (5.35.249.60) 56(84) bytes of data.
```

2.3 Viewers

In order to create a 'fake' viewer (for pdf or img), environments are:

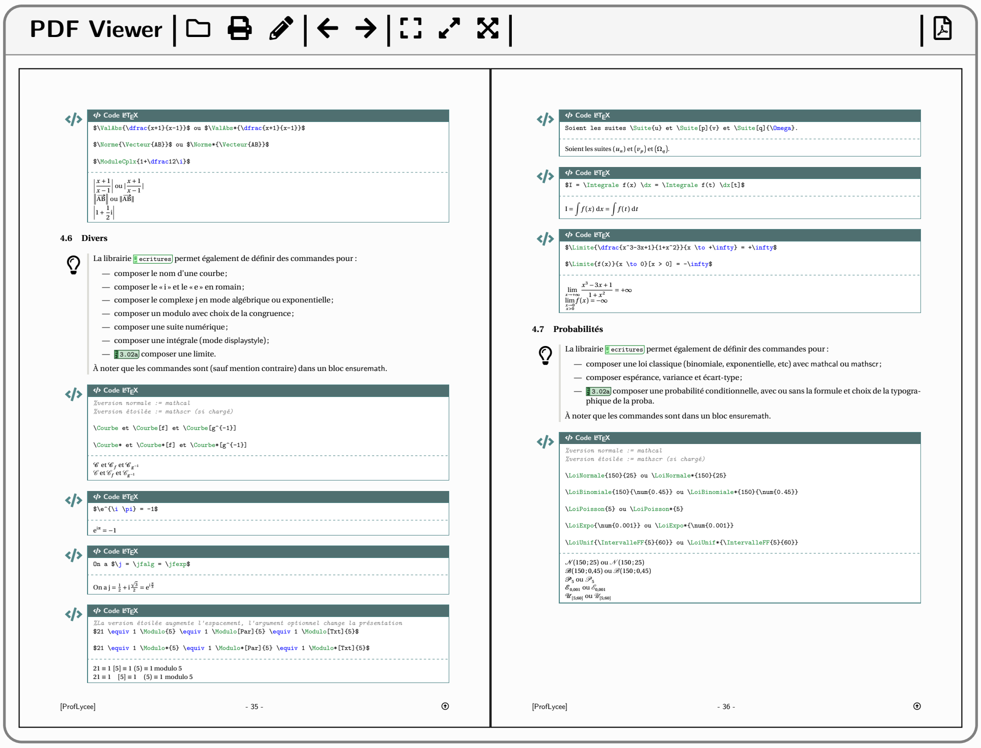
```
%----PDF Viewer like
\begin{PDFViewer}[keys]{tcbox options}
    ....
\end{PDFViewer}

%----Image Viewer like
\begin{IMGViewer}[keys]{tcbox options}
    ....
\end{IMGViewer}

%----Script editor like
\begin{PYViewer}[keys]{tcbox options}
    ....
\end{PYViewer}
```

Optional keys, between `[...]` are:

- `Title`: title of the viewer;
- `Align`: horizontal alignment of the box (`center` by default);
- `Width`: width of the box (`\linewidth` by default);
- `Halign`: horizontal alignment fot the content (`left` by default);
- `Icons`: boolean for the icons (`true` by default).

The mandatory argument, between `{...}`, are options to give to the tcbox.

```
\begin{PDFViewer}{hbox}
\fbox{\includegraphics[page=35,width=6cm]{ProfLycee-doc.pdf}}%
\fbox{\includegraphics[page=36,width=6cm]{ProfLycee-doc.pdf}}
\end{PDFViewer}
```



Code LaTeX

```
$\ValAbs{\dfrac{x+1}{x-1}}$ ou $\ValAbs*{\dfrac{x+1}{x-1}}$

$\Norme{\Vecteur{AB}}$ ou $\Norme*{\Vecteur{AB}}$

$\ModuleCplx{1+\dfrac12\i}$
```

$\left|\dfrac{x+1}{x-1}\right|$ ou $|\dfrac{x+1}{x-1}|$
$\left\|\overrightarrow{AB}\right\|$ ou $\|\overrightarrow{AB}\|$
$\left|1+\dfrac{1}{2}\mathrm{i}\right|$

4.6 Divers

La librairie ecritures permet également de définir des commandes pour :

- composer le nom d'une courbe ;
- composer le « i » et le « e » en romain ;
- composer le complexe j en mode algébrique ou exponentielle ;
- composer un modulo avec choix de la congruence ;
- composer une suite numérique ;
- composer une intégrale (mode displaystyle) ;
- 3.02a composer une limite.

À noter que les commandes sont (sauf mention contraire) dans un bloc ensuremath.

Code LaTeX

```
%version normale := mathcal
%version étoilée := mathscr (si chargé)

\Courbe et \Courbe[f] et \Courbe[g^{-1}]

\Courbe* et \Courbe*[f] et \Courbe*[g^{-1}]
```

$\mathcal{C}$ et $\mathcal{C}_f$ et $\mathcal{C}_{g^{-1}}$
$\mathscr{C}$ et $\mathscr{C}_f$ et $\mathscr{C}_{g^{-1}}$

Code LaTeX

```
$\e^{\i \pi} = -1$
```

$\mathrm{e}^{\mathrm{i}\pi} = -1$

Code LaTeX

```
On a $\j = \jfalg = \jfexp$
```

On a $\mathrm{j} = \frac{1}{2} + \mathrm{i}\frac{\sqrt{3}}{2} = \mathrm{e}^{\mathrm{i}\frac{\pi}{3}}$

Code LaTeX

```
%La version étoilée augmente l'espacement, l'argument optionnel change la présentation
$21 \equiv 1 \Modulo{5} \equiv 1 \Modulo[Par]{5} \equiv 1 \Modulo[Txt]{5}$

$21 \equiv 1 \Modulo*{5} \equiv 1 \Modulo*[Par]{5} \equiv 1 \Modulo*[Txt]{5}$
```

$21 \equiv 1\ [5] \equiv 1\ (5) \equiv 1$ modulo 5
$21 \equiv 1 \quad [5] \equiv 1 \quad (5) \equiv 1$ modulo 5

[ProfLycee] - 35 -

Code LaTeX

```
Soient les suites \Suite{u} et \Suite[p]{v} et \Suite[q]{\Omega}.
```

Soient les suites (u_n) et (v_p) et (Ω_q).

Code LaTeX

```
$I = \Integrale f(x) \dx = \Integrale f(t) \dx[t]$
```

$\mathrm{I} = \displaystyle\int f(x)\,\mathrm{d}x = \int f(t)\,\mathrm{d}t$

Code LaTeX

```
$\Limite{\dfrac{x^3-3x+1}{1+x^2}}{x \to +\infty} = +\infty$

$\Limite{f(x)}{x \to 0}[x > 0] = -\infty$
```

$\lim\limits_{x\to+\infty} \dfrac{x^3-3x+1}{1+x^2} = +\infty$
$\lim\limits_{\substack{x\to 0\\ x>0}} f(x) = -\infty$

4.7 Probabilités

La librairie ecritures permet également de définir des commandes pour :

- composer une loi classique (binomiale, exponentielle, etc) avec mathcal ou mathscr ;
- composer espérance, variance et écart-type ;
- 3.02a composer une probabilité conditionnelle, avec ou sans la formule et choix de la typographique de la proba.

À noter que les commandes sont dans un bloc ensuremath.

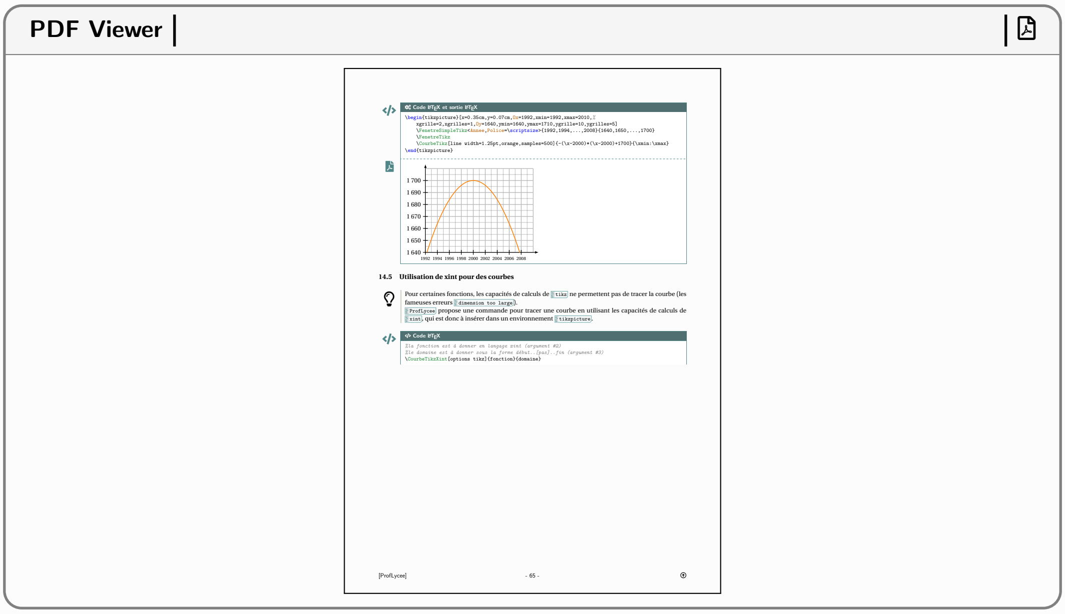
Code LaTeX

```
%version normale := mathcal
%version étoilée := mathscr (si chargé)

\LoiNormale{150}{25} ou \LoiNormale*{150}{25}

\LoiBinomiale{150}{\num{0.45}} ou \LoiBinomiale*{150}{\num{0.45}}

\LoiPoisson{5} ou \LoiPoisson*{5}

\LoiExpo{\num{0.001}} ou \LoiExpo*{\num{0.001}}

\LoiUnif{\IntervalleFF{5}{60}} ou \LoiUnif*{\IntervalleFF{5}{60}}
```

$\mathcal{N}(150\,;25)$ ou $\mathscr{N}(150\,;25)$
$\mathcal{B}(150\,;0{,}45)$ ou $\mathscr{B}(150\,;0{,}45)$
$\mathcal{P}_5$ ou $\mathscr{P}_5$
$\mathcal{E}_{0{,}001}$ ou $\mathscr{E}_{0{,}001}$
$\mathcal{U}_{[5;60]}$ ou $\mathscr{U}_{[5;60]}$

[ProfLycee] - 36 -

```
\begin{PDFViewer}[Width=14cm,Icons=false]{}
\fbox{\includegraphics[page=65,width=4.75cm]{ProfLycee-doc.pdf}}
\end{PDFViewer}
```



Code LaTeX et sortie LaTeX

```
\begin{tikzpicture}[x=0.35cm,y=0.07cm,Ox=1992,xmin=1992,xmax=2010,%
    xgrille=2,xgrilles=1,Oy=1640,ymin=1640,ymax=1710,ygrille=10,ygrilles=5]
    \FenetreSimpleTikz<Annee,Police=\scriptsize>{1992,1994,...,2008}{1640,1650,...,1700}
    \FenetreTikz
    \CourbeTikz[line width=1.25pt,orange,samples=500]{-(\x-2000)*(\x-2000)+1700}{\xmin:\xmax}
\end{tikzpicture}
```

14.5 Utilisation de xint pour des courbes

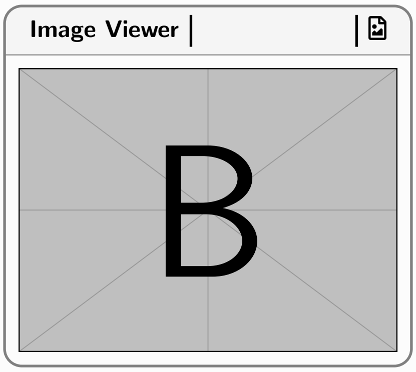
Pour certaines fonctions, les capacités de calculs de tikz ne permettent pas de tracer la courbe (les fameuses erreurs dimension too large).
ProfLycee propose une commande pour tracer une courbe en utilisant les capacités de calculs de xint, qui est donc à insérer dans un environnement tikzpicture.

Code LaTeX

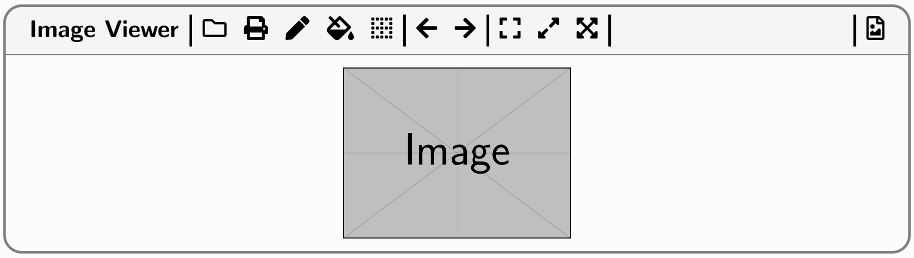
```
%la fonction est à donner en langage xint (argument #2)
%le domaine est à donner sous la forme début..[pas]..fin (argument #3)
\CourbeTikzXint[options tikz]{fonction}{domaine}
```

[ProfLycee] - 65 -

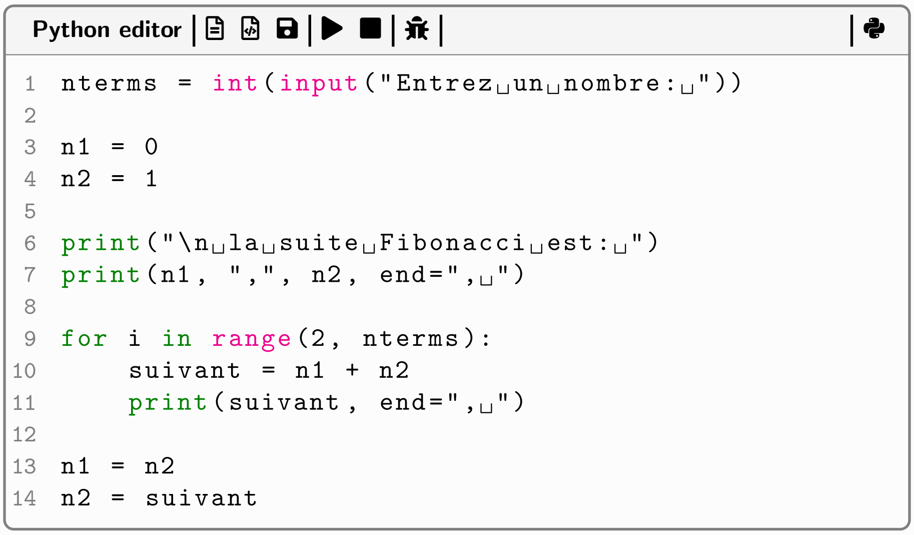
```
\begin{IMGViewer}[Icons=false]{hbox}
\includegraphics[width=5cm]{example-image-b}
\end{IMGViewer}
```



```
\begin{IMGViewer}[Width=12cm]{}
\includegraphics[width=3cm]{example-image}
\end{IMGViewer}
```



```
%with listings, or piton, for example
\begin{PYViewer}[width=12cm]{}
\begin{lstlisting}%
    [
        language=python,basicstyle=\ttfamily\small,
        keywordstyle=\color{green!50!black},tabsize=4,
        keywordstyle={[2]\color{magenta}},
        numbers=left,numbersep=3mm,xleftmargin=5mm,
        aboveskip=0pt,belowskip=0pt,
        numberstyle=\footnotesize\ttfamily\color{gray}
    ]
nterms = int(input("Entrez un nombre: "))

n1 = 0
n2 = 1

print("\n la suite Fibonacci est: ")
print(n1, ",", n2, end=", ")

for i in range(2, nterms):
    suivant = n1 + n2
    print(suivant, end=", ")

n1 = n2
n2 = suivant
\end{lstlisting}
\end{PYViewer}
```



```
nterms = int(input("Entrez␣un␣nombre:␣"))

n1 = 0
n2 = 1

print("\n␣la␣suite␣Fibonacci␣est:␣")
print(n1, ",", n2, end=",␣")

for i in range(2, nterms):
    suivant = n1 + n2
    print(suivant, end=",␣")

n1 = n2
n2 = suivant
```

2.4 Folders/files as in explorer

The `forest` package, with his library `edges`, can present foldertrees.

```
\begin{ForestDirTree}[keys]{forest options}
    [folder,FTdir
        [subfolder1,FTdir]
        [subfolder2,FTdir
            [file1,FTfile]
            [file2,FTfile]
        ...
    ]
        ...
    ]
\end{ForestDirTree}
```

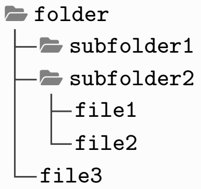
Available keys, betweeen `[...]`, are:

- `font`: font of texts;
- `coliconfolder`: `gray` by default;
- `coliconfile`: `gray` by default;
- `iconfolders`: boolean for folder icons;
- `iconfiles`: boolean for file icons;
- `vsep`: vertical space between items (`0.15em` by default);
- `iconfolder`: icon for folder;
- `iconfile`: icone for file.

Folders need to be given within `<folder_name>,FTdir`.
Files need to be given within `<file_name>,FTfile`.

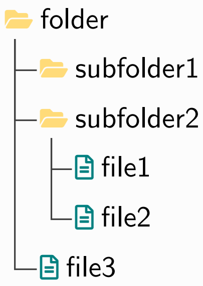
Mandatory argument, between `{...}`, corresponds to forest specific commands.

```
\begin{ForestDirTree}{}
    [folder,FTdir
        [subfolder1,FTdir]
        [subfolder2,FTdir
            [file1,FTfile]
            [file2,FTfile]
        ]
        [file3,FTfile]
    ]
\end{ForestDirTree}
```



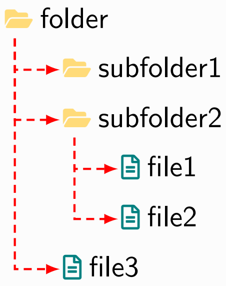
```
folder
├─subfolder1
├─subfolder2
│ ├─file1
│ └─file2
└─file3
```

```
\begin{ForestDirTree}%

    [font=\sffamily,coliconfolder=yellow!50!pink,iconfiles,coliconfile=teal,vsep=0.5em]%
    {}
    [folder,FTdir
        [subfolder1,FTdir]
        [subfolder2,FTdir
            [file1,FTfile]
            [file2,FTfile]
        ]
        [file3,FTfile]
    ]
\end{ForestDirTree}
```



Styles are global, but can be locally defined.

```
\tikzset{FTedge/.style={thick,->,densely dashed,red,>=latex}}
\begin{ForestDirTree}%

    [font=\sffamily,coliconfolder=yellow!50!pink,iconfiles,coliconfile=teal,vsep=0.5em]%
    {l sep=2em}
    [folder,FTdir
        [subfolder1,FTdir]
        [subfolder2,FTdir
            [file1,FTfile]
            [file2,FTfile]
        ]
        [file3,FTfile]
    ]
\end{ForestDirTree}
```



2.5 Posts as in Mastodon or BlueSky

It’s possible to present posts as in Mastodon or BlueSky.

```
\begin{MastodonPost}[options]<tcolorbox keys>
...
\end{MastodonPost}

\begin{BlueSkyPost}[options]<tcolorbox keys>
...
\end{BlueSkyPost}
```

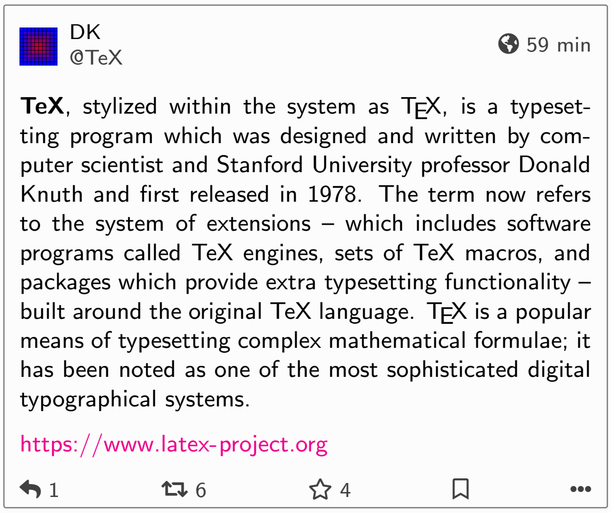
Available keys, betweeen `[...]`, are:

- `width`: width of box;
- `avatar`: image of the avatar (square if possible !);
- `name`: pseudo of the account
- `account`: account name;
- `time`: time of post;
- `counters`: counters for views/likes/shares;
- `font`: font fir main text;
- `colbg`: background color of box.

Headers and footers are given by following macros, and can be redefined if necessary !

```
\newcommand\mastodonheader{%
        \begin{tblr}{width=\linewidth,colspec={Q[c,m]X[m,l]Q[m,l]},
                colsep=0pt,cells={font=\scriptsize\sffamily}}

    $\vcenter{\hbox{\includegraphics[height=4ex]{\mastodonaccountavatar}}}$&
    {\hspace*{1.25ex}\mastodonaccountname \\
    \hspace*{1.25ex}\color{darkgray}{\mastodonaccountadress}} &
    \color{darkgray}{\faGlobeAmericas~\mastodontime} \\
        \end{tblr}\par\medskip
}

\newcommand\blueskyheader{%
        {\scriptsize\sffamily\textbf{\mastodonaccountname}~
        \textcolor{darkgray}{\mastodonaccountadress~
        \textperiodcentered~\mastodontime}}\par\medskip
}

\newcommand\mastodonfooter{%

    \textcolor{darkgray}{\scriptsize\scalebox{-1}[1]{\faShare}~\mastodonnumbers[1]
    \hfill \faRetweet~\mastodonnumbers[2] \hfill \faStar[regular]~\mastodonnumbers[3]
    \hfill \faBookmark[regular] \hfill \faEllipsisH}
}

\newcommand\blueskyfooter{%
        \textcolor{darkgray}{\scriptsize\faComment*[regular]~\mastodonnumbers[1]
    \hfill \faRetweet~\mastodonnumbers[2] \hfill
    \faHeart[regular]~\mastodonnumbers[3] \hfill \faEllipsisH \hfill~}
}
```

```
\begin{MastodonPost}
\textbf{TeX}, stylized within the system as \TeX, is a typesetting program which was
    designed and written by computer scientist and Stanford University professor
    Donald Knuth and first released in 1978. The term now refers to the system of
    extensions  which includes software programs called TeX engines, sets of TeX
    macros, and packages which provide extra typesetting functionality  built around
    the original TeX language. \TeX\ is a popular means of typesetting complex
    mathematical formulae; it has been noted as one of the most sophisticated digital
    typographical systems.

\medskip

\url{https://www.latex-project.org}
\end{MastodonPost}
```



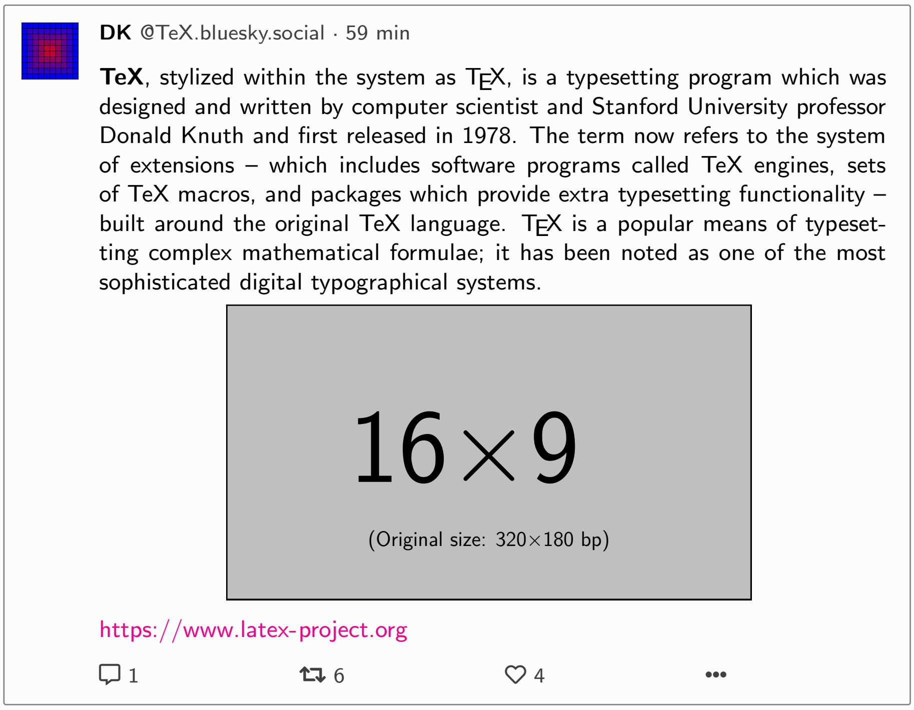
DK
@TeX

🌐 59 min

TeX, stylized within the system as TeX, is a typesetting program which was designed and written by computer scientist and Stanford University professor Donald Knuth and first released in 1978. The term now refers to the system of extensions – which includes software programs called TeX engines, sets of TeX macros, and packages which provide extra typesetting functionality – built around the original TeX language. TeX is a popular means of typesetting complex mathematical formulae; it has been noted as one of the most sophisticated digital typographical systems.

https://www.latex-project.org

```
\begin{BlueSkyPost}[width=12cm,account={\symbol{64}TeX.bluesky.social}]
\textbf{TeX}, stylized within the system as \TeX, is a typesetting program which was
    designed and written by computer scientist and Stanford University professor
    Donald Knuth and first released in 1978. The term now refers to the system of
    extensions  which includes software programs called TeX engines, sets of TeX
    macros, and packages which provide extra typesetting functionality  built around
    the original TeX language. \TeX\ is a popular means of typesetting complex
    mathematical formulae; it has been noted as one of the most sophisticated digital
    typographical systems.

\smallskip

\hfill\includegraphics[width=0.667\linewidth]{example-image-16x9.pdf}\hfill~

\smallskip

\url{https://www.latex-project.org}
\end{BlueSkyPost}
```



DK @TeX.bluesky.social · 59 min

TeX, stylized within the system as TEX, is a typesetting program which was designed and written by computer scientist and Stanford University professor Donald Knuth and first released in 1978. The term now refers to the system of extensions – which includes software programs called TeX engines, sets of TeX macros, and packages which provide extra typesetting functionality – built around the original TeX language. TEX is a popular means of typesetting complex mathematical formulae; it has been noted as one of the most sophisticated digital typographical systems.

https://www.latex-project.org

🗨 1 ⟲ 6 ♡ 4 •••