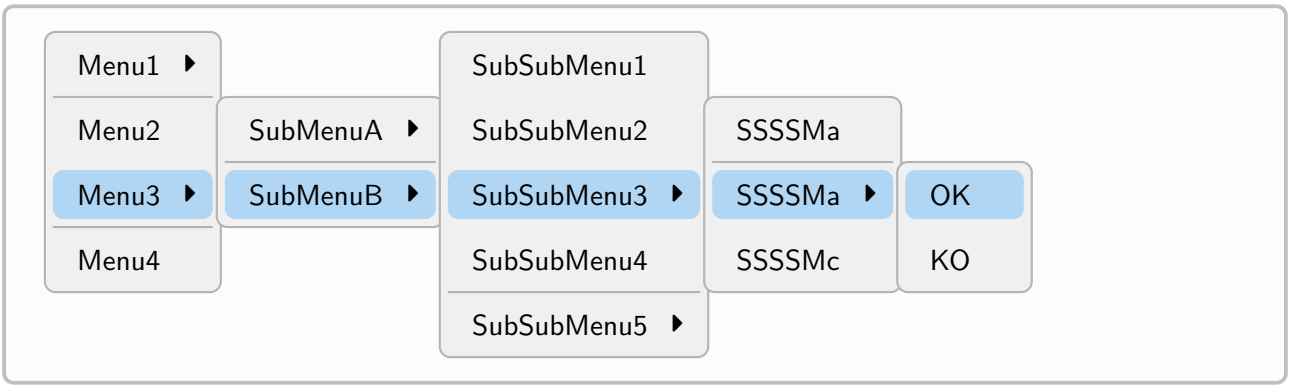
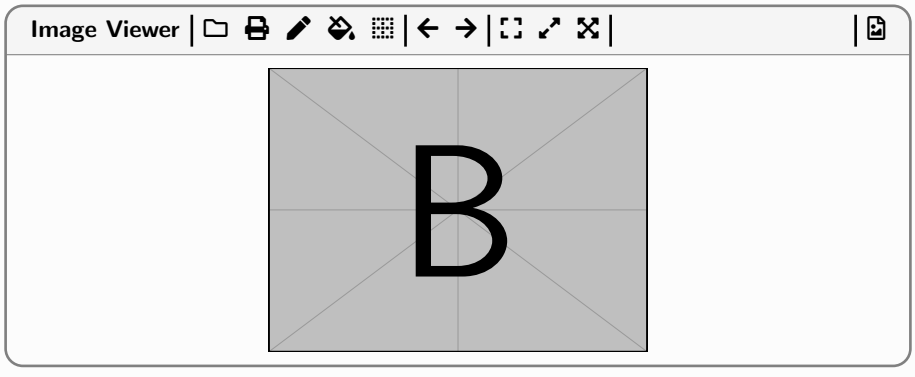
sim-os-menus

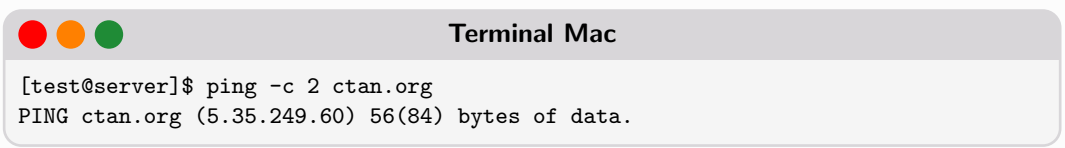
Simulate 'windows', 'terminal' or 'context menu' like in an OS.

Version 0.1.3 -- 28/09/2024

Cédric Pierquet
c pierquet -- at -- outlook . fr
https://github.com/cpierquet/sim-os-menus







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`).

1.3 History

```
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,... § ...`

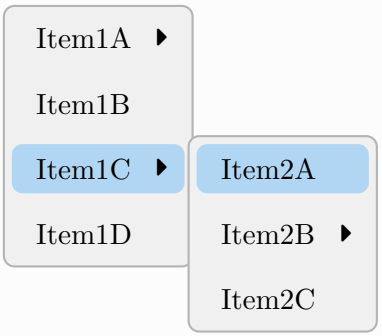
- if an item ends with `(*)`, this is the beginning of the next level (only one by level !);
- if an item ends with `(>)` (before optional `(*)`), an arrow is written at the end of the line.

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

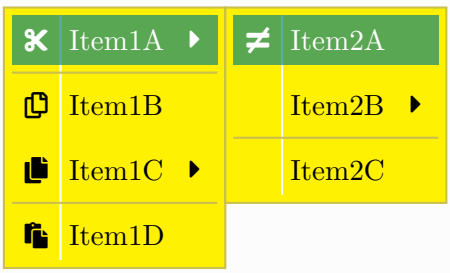
A few tips, due to `ListIcons`, `ListOffsets` and `ListSeps` keys, which are *sensitive*:

- `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 !).

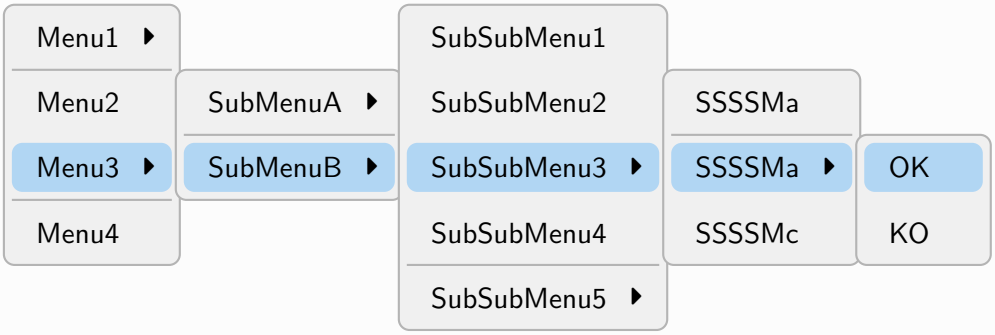
```
%default style
\ContextMenu{Item1A(>),Item1B,Item1C(>)(*),Item1D § Item2A(*),Item2B(>),Item2C}
```




```
%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}
```

Terminal Win

```
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}
```

Terminal UNiX

```
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}
```

Terminal Mac

```
[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}
```

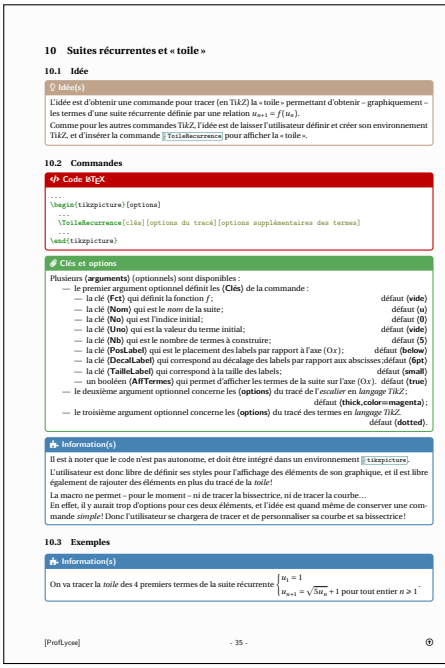
Optional keys, between `[...]` are:

- `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}
```

PDF Viewer



10 Suites récurrentes et « toile »

10.1 Idée

Idée(s)

L'idée est d'obtenir une commande pour tracer (en TikZ) la « toile » permettant d'obtenir – graphiquement – les termes d'une suite récurrente définie par une relation $u_{n+1} = f(u_n)$.
Comme pour les autres commandes TikZ, l'idée est de laisser l'utilisateur définir et créer son environnement TikZ, et d'insérer la commande `\ToileRecurrence` pour afficher la « toile ».

10.2 Commandes

Code LaTeX

```
...
\begin{tikzpicture}[options]
  ...
  \ToileRecurrence[clés][options du tracé][options supplémentaires des termes]
  ...
\end{tikzpicture}
```

Clés et options

Plusieurs (**arguments**) (optionnels) sont disponibles :
- le premier argument optionnel définit les (**Clés**) de la commande :
  - la clé (**Fct**) qui définit la fonction f ; défaut (**vide**)
  - la clé (**Nom**) qui est le *nom* de la suite ; défaut (**u**)
  - la clé (**No**) qui est l'indice initial ; défaut (**0**)
  - la clé (**Uno**) qui est la valeur du terme initial ; défaut (**vide**)
  - la clé (**Nb**) qui est le nombre de termes à construire ; défaut (**5**)
  - la clé (**PosLabel**) qui est le placement des labels par rapport à l'axe (Ox) ; défaut (**below**)
  - la clé (**DecalLabel**) qui correspond au décalage des labels par rapport aux abscisses ; défaut (**6pt**)
  - la clé (**TailleLabel**) qui correspond à la taille des labels ; défaut (**small**)
  - un booléen (**AffTermes**) qui permet d'afficher les termes de la suite sur l'axe (Ox). défaut (**true**)
- le deuxième argument optionnel concerne les (**options**) du tracé de l'*escalier* en *langage TikZ* ; défaut (**thick,color=magenta**) ;
- le troisième argument optionnel concerne les (**options**) du tracé des termes en *langage TikZ*. défaut (**dotted**).

Information(s)

Il est à noter que le code n'est pas autonome, et doit être intégré dans un environnement `tikzpicture`.
L'utilisateur est donc libre de définir ses styles pour l'affichage des éléments de son graphique, et il est libre également de rajouter des éléments en plus du tracé de la *toile*!
La macro ne permet – pour le moment – ni de tracer la bissectrice, ni de tracer la courbe…
En effet, il y aurait trop d'options pour ces deux éléments, et l'idée est quand même de conserver une commande *simple*! Donc l'utilisateur se chargera de tracer et de personnaliser sa courbe et sa bissectrice!

10.3 Exemples

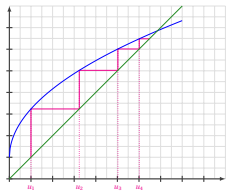
Information(s)

On va tracer la *toile* des 4 premiers termes de la suite récurrente $\begin{cases} u_1 = 1 \\ u_{n+1} = \sqrt{5u_n} + 1 \text{ pour tout entier } n \geqslant 1 \end{cases}$

[ProfLycee] - 35 -

Code LaTeX

```
%code tikz
\def\y{1.5cm}\def\y{1.5cm}
\def\xmin{0}\def\xmax{10}\def\xgrille{1}\def\xgrilles{0.5}
\def\ymin{0}\def\ymax{8}\def\ygrille{1}\def\ygrilles{0.5}
%axes et grilles
\draw[xstep=\xgrilles,ystep=\ygrilles,line width=0.6pt,lightgray!50] (\xmin,\ymin) grid
  (\xmax,\ymax);
\draw[line width=1.5pt,->,darkgray,>=latex] (\xmin,0)--(\xmax,0) ;
\draw[line width=1.5pt,->,darkgray,>=latex] (0,\ymin)--(0,\ymax) ;
\foreach \x in {0,1,...,9} {\draw[darkgray,line width=1.5pt] (\x,4pt) -- (\x,-4pt) ;}
\foreach \y in {0,1,...,7} {\draw[darkgray,line width=1.5pt] (4pt,\y) -- (-4pt,\y) ;}
%fonctions définie et réutilisable
\def\f{sqrt(5*\x)+1}
%toile
\ToileRecurrence[Fct={\f},No=1,Uno=1,Nb=4,DecalLabel=4pt]
%éléments supplémentaires
\draw[very thick,blue,domain=0:8,samples=250] plot (\x,{\f}) ;
\draw[very thick,ForestGreen,domain=0:8,samples=2] plot (\x,\x) ;
```



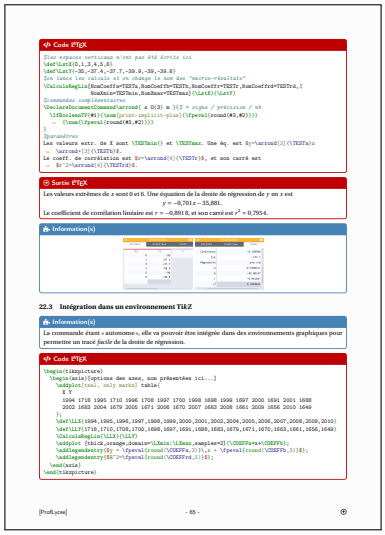
Information(s)

Peut-être que – ultérieurement – des options *booléennes* seront disponibles pour un tracé *générique* de la courbe et de la bissectrice, mais pour le moment la macro ne fait *que* l'escalier.

[ProfLycee] - 36 -

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

PDF Viewer



Code LaTeX

```
%Les espaces verticaux n'ont pas été écrits ici
\def\Listx{0,1,3,4,5,6}
\def\Listy{-35,-37.4,-37.7,-39.9,-39,-39.6}
%On lance les calculs et on change le nom des "macros-résultats"
\CalculeReglin[NomCoeffa=TESTa,NomCoeffb=TESTb,NomCoeffr=TESTr,NomCoeffrd=TESTrd,%
               NomAire=TESTair,NomAmax=TESTmax]{\Listx}{\Listy}
%commandes complémentaires
\DeclareDocumentCommand\arrond{ m O{3} m }{{ % signe / précision / nb
  \IfBooleanTF{#1}{\num[print-implicit-plus]{\fpeval{round(#3,#2)}}}
    {\num{\fpeval{round(#3,#2)}}}
}
%paramètres
Les valeurs extr. de X sont \TESTmin{} et \TESTmax. Une éq. est $y=\arrond{3}{\TESTa}x
  -\arrond{3}{\TESTb}$.
Le coeff. de corrélation est $r=\arrond{4}{\TESTr}$, et son carré est
  $r^2=\arrond{4}{\TESTrd}$.
```

Sortie LaTeX

Les valeurs extrêmes de x sont 0 et 6. Une équation de la droite de régression de y en x est
$y = -6{,}701\,x - 35{,}891$.
Le coefficient de corrélation linéaire est $r = -0{,}8918$, et son carré est $r^2 = 0{,}7954$.

Information(s)

22.3 Intégration dans un environnement TikZ

Information(s)

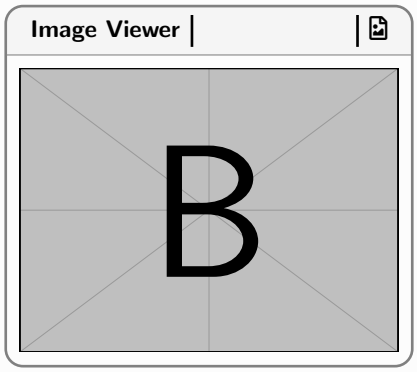
La commande étant « autonome », elle va pouvoir être intégrée dans des environnements graphiques pour permettre un tracé facile de la droite de régression.

Code LaTeX

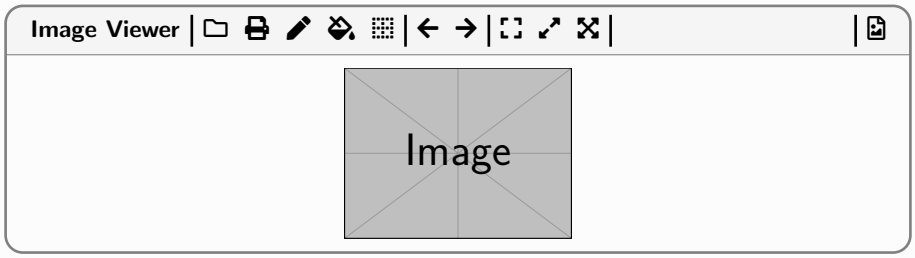
```
\begin{tikzpicture}
  \begin{axis}[options des axes, non présentées ici...]
    \addplot[teal, only marks] table{
      X Y
      1994 1718 1995 1710 1996 1708 1997 1700 1998 1698 1999 1697 2000 1691 2001 1688
      2002 1683 2003 1679 2005 1671 2006 1670 2007 1663 2008 1661 2009 1656 2010 1649
    };
    \def\LLX{1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2005,2006,2007,2008,2009,2010}
    \def\LLY{1718,1710,1708,1700,1698,1697,1691,1688,1683,1679,1671,1670,1663,1661,1656,1649}
    \CalculeReglin{\LLX}{\LLY}
    \addplot [thick,orange,domain=\LXmin:\LXmax,samples=2]{\COEFFa*x+\COEFFb};
    \addlegendentry{$y = \fpeval{round(\COEFFa,3)}\,x + \fpeval{round(\COEFFb,3)}$};
    \addlegendentry{$R^2=\fpeval{round(\COEFFrd,3)}$};
  \end{axis}
\end{tikzpicture}
```

[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}
```

Python editor

```
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 explrer

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}
```

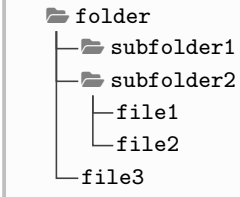
Available keys, betweeen `[...]`, are:

- `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`.

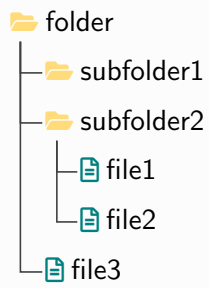
Mandatory argument, between `{...}`, corresponds to forest specific commands.

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



```
\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}
```