

Python et L^AT_EX

Le langage Python a été choisi par l'éducation nationale pour être enseigné au niveau lycée ; on aurait pu lui préférer un langage qui nécessite la définition et le typage des variables. Tant pis ! On va donc avoir à écrire des listings de programmes en Python, comme dans l'épreuve de CAPES maths option informatique 2018 disponible [ici](#) grâce à la diligence de Denis VERGÈS.

1 Les bases

Les listings sont composés à partir de l'extension `listings` que l'on chargera en entrant dans le préambule : `\usepackage{listings}`.

Cette extension est directement disponible avec les systèmes MiK_TE_X et T_EXLive.

Le principe : on indique à l'extension `listings` quel est le langage avec lequel on travaille et l'extension en reconnaît les mots-clés qu'il peut ainsi mettre en évidence selon le choix de l'utilisateur. Voyons le « réglage de base » pour Python : ces renseignements ont été extraits du fichier `lstdrvrs.pdf` disponible [ici](#). On trouve le même paramétrage dans le fichier `listings-python.prf`, fichier des préférences de Python.

```
\definecolor{purple2}{RGB}{153,0,153} % there's actually no standard purple
\definecolor{green2}{RGB}{0,153,0} % a darker green

\lstset{%
  language=Python,                % the language
  basicstyle=\normalsize\ttfamily, % size of the fonts for the code
  % Color settings to match IDLE style
  keywordstyle=\color{orange},     % core keywords
  keywordstyle={[2]\color{purple2}}, % built-ins
  stringstyle=\color{green2},
  commentstyle=\color{red},
  upquote=true,                   % requires textcomp
}
```

Cette définition est simple à comprendre : on indique le langage du programme dont on veut le listing, on définit la police de caractères du listing, puis on définit les différentes couleurs des mots-clés, des chaînes et des commentaires (nécessite l'extension `xcolor`), enfin on indique le style des apostrophes (nécessite l'extension `textcomp`).

2 Premier listing

Voici comment on écrit un listing Python :

```
\begin{lstlisting}
# CAPES externe 2018

from math import *

def lucas1(n):
    if n == 0:
        return 2
    phi = (1+sqrt(5))/2
    phi2 = (1-sqrt(5))/2
    return phi**n + phi2**n
\end{lstlisting}
```

Cela donne avec les paramètres précédemment définis :

```
# CAPES externe 2018

from math import *

def lucas1(n):
    if n == 0:
        return 2
    phi = (1+sqrt(5))/2
    phi2 = (1-sqrt(5))/2
    return phi**n + phi2**n
```

Les indentations se font **exclusivement** avec la touche de tabulation ; on verra plus loin que l'on peut visualiser ces tabulations.

On voit que les mots-clés ainsi que le commentaire ont été reconnus et mis dans la couleur souhaitée.

3 Un peu mieux

On peut agrémenter un peu ce listing par quelques commandes ; voyons celui-ci qui est encadré et dont les lignes sont numérotées :

```
0 # CAPES externe 2018
1
2 from math import *
3
4 def lucas1(n):
5     if n == 0:
6         return 2
7     phi = (1+sqrt(5))/2
8     phi2 = (1-sqrt(5))/2
9     return phi**n + phi2**n
```

Différents paramètres ont été rajoutés dans une deuxième liste :

```
\lstset{%
frame=single,%           style du cadre - none pour ne pas encadrer
numbers=left,%          numéros des lignes à gauche - none pour ne pas numéroter
numberstyle=\small,%   on peut mettre \tiny ou \normalsize (taille par défaut)
firstnumber=0,%        numéro de la première ligne
tabsize=6,%            largeur des tabulations, 8 par défaut
showtabs=false,%       permet de voir les tabulations quand true
xleftmargin=20pt,%     marge gauche
xrightmargin=0pt,%    marge droite qui peut rester à 0
framexleftmargin=0pt,% pour inclure les numéros dans le cadre : mettre 20pt
}
```

Les commentaires me paraissent clairs, mais on a toutes les options possibles dans le document [listings.pdf](#).

Ces options s'ajoutent aux précédentes et restent opérationnelles tant qu'on n'en a pas défini d'autres.

4 Pour photocopier

Comme il est rare que l'on photocopie en couleurs, on peut remplacer les couleurs par du gras ou de l'italique ; il suffit de redéfinir les options en encapsulant le tout entre deux accolades pour ne pas perturber les listings suivants.

J'en profite pour montrer un listing avec des numéros de lignes inclus dans le cadre.

```
0 # CAPES externe 2018
1
2 from math import *
3
4 def lucas1(n):
5     if n == 0:
6         return 2
7     phi = (1+sqrt(5))/2
8     phi2 = (1-sqrt(5))/2
9     return phi**n + phi2**n
```

On l'obtient par :

```
{% début encapsulage
\lstset{%
    keywordstyle=\pmb,% "gras du pauvre"
    stringstyle=\pmb,
    commentstyle=\itshape,% italique
    framexleftmargin=20pt,% numéros des lignes dans le cadre
}% fin lstset
\begin{lstlisting}
...
\end{lstlisting}
}% fin encapsulage
```

Une petite remarque à propos du **gras** : la fonte utilisée dans les listings, style « machine à écrire », ne connaît pas le mode **gras** ; il faut donc le simuler avec `\pmb` qui signifie « poor man bold » ! Une autre solution consiste à utiliser une autre police de caractères pour les listings en en choisissant une qui connaît le mode **gras**.

5 Fond de couleur

Les amateurs de couleur pourront produire des listings sur fond coloré en définissant l'option `backgroundcolor` comme dans `backgroundcolor=\color{yellow!30}`, par exemple :

```
0 # CAPES externe 2018
1
2 from math import *
3
4 def lucas1(n):
5     if n == 0:
6         return 2
7     phi = (1+sqrt(5))/2
8     phi2 = (1-sqrt(5))/2
9     return phi**n + phi2**n
```

6 Listing partiel

Enfin on peut n'afficher que quelques lignes d'un listing complet en définissant `firstline` et `lastline` en option (la numérotation commençant à 1) :

```
0 def lucas1(n):
1     if n == 0:
2         return 2
3     phi = (1+sqrt(5))/2
4     phi2 = (1-sqrt(5))/2
5         return phi**n + phi2**n
```

On l'obtient par :

```
\begin{lstlisting}[firstline=5,lastline=10]
# CAPES externe 2018

from math import *

def lucas1(n):
    if n == 0:
        return 2
    phi = (1+sqrt(5))/2
    phi2 = (1-sqrt(5))/2
    return phi**n + phi2**n
\end{lstlisting}
```

Attention toutefois à la numérotation des lignes ; il faudra l'ajuster « à la main » si on veut qu'elle corresponde à celle des listings précédents :

```
4 def lucas1(n):
5     if n == 0:
6         return 2
7     phi = (1+sqrt(5))/2
8     phi2 = (1-sqrt(5))/2
9         return phi**n + phi2**n
```

On l'obtient en entrant :

```
{
\lstset{firstnumber=4}
\begin{lstlisting}[firstline=5,lastline=10]
# CAPES externe 2018

from math import *

def lucas1(n):
    if n == 0:
        return 2
    phi = (1+sqrt(5))/2
    phi2 = (1-sqrt(5))/2
    return phi**n + phi2**n
\end{lstlisting}
}
```

Bon Python!

□