

# ProfLycee (3.01f), Piton et Pyluatex

## 1 Code « Piton », indépendant de Pyluatex

### 1.1 Préambule basique

```
\documentclass[french,a4paper,10pt]{article}
\usepackage{ProfLycee}
\useproflyclib{piton} % lua
```

### 1.2 Exemples

```
%Sortie par défaut
\begin{CodePiton}{}
def valeur_absolue(x):
    "Renvoie la valeur absolue de x"
    #le petit test qui va bien
    if x > 0:
        return x
    else:
        return -x
\end{CodePiton}
```

</> Code Python

```
1 def valeur_absolue(x):
2     "Renvoie la valeur absolue de x"
3     #le petit test qui va bien
4     if x > 0:
5         return x
6     else:
7         return -x
```

```
%Sortie avec style Classique, Largeur=10cm
\begin{CodePiton}[Largeur=10cm]{}
def valeur_absolue(x):
    "Renvoie la valeur absolue de x"
    #le petit test qui va bien
    if x > 0:
        return x
    else:
        return -x
\end{CodePiton}
```

</> Code Python

```
1 def valeur_absolue(x):
2     "Renvoie la valeur absolue de x"
3     #le petit test qui va bien
4     if x > 0:
5         return x
6     else:
7         return -x
```

```
%Sortie avec Style=Moderne, Sans Titre, Largeur=10cm, centré
\begin{CodePiton}[Style=Moderne,Largeur=10cm,BarreTitre=false,Alignement=center]{}
def valeur_absolue(x):
    "Renvoie la valeur absolue de x"
    #le petit test qui va bien
    if x > 0:
        return x
    else:
        return -x
\end{CodePiton}
```

Code Python

```
1 def valeur_absolue(x):
2     "Renvoie la valeur absolue de x"
3     #le petit test qui va bien
4     if x > 0:
5         return x
6     else:
7         return -x
```

```
%Sortie avec Style=Classique, Largeur=0.5\linewidth, aligné à droite, sans Cadre, avec Filigrane
\begin{CodePiton}%
    [Largeur=0.5\linewidth,Cadre=false,Alignement=flush right,Filigrane,Titre={Script}]{}
#environnement piton avec numéros de ligne, pleine largeur, style moderne
def valeur_absolue(x):
    "Renvoie la valeur absolue de x"
    #le petit test qui va bien
    if x > 0:
        return x
    else:
        return -x
\end{CodePiton}
```

Script

```
1 def valeur_absolue(x):
2     "Renvoie la valeur absolue de x"
3     #le petit test qui va bien
4     if x > 0:
5         return x
6     else:
7         return -x
```

```
%Sortie Moderne, Largeur=11cm, avec Filigrane, aligné à gauche, sans ligne
\begin{CodePiton}[Style=Moderne,Largeur=11cm,Filigrane,Alignement=flush left,Lignes=false]{}
def valeur_absolue(x):
    "Renvoie la valeur absolue de x"
    #le petit test qui va bien
    if x > 0:
        return x
    else:
        return -x
\end{CodePiton}
```

Code Python

```
def valeur_absolue(x):
    "Renvoie la valeur absolue de x"
    #le petit test qui va bien
    if x > 0:
        return x
    else:
        return -x
```

## 2 Console « Piton », dépendant de Pyluatex

### 2.1 Préambule, avec le package pyluatex

```
\documentclass[french,a4paper,10pt]{article}
\usepackage{ProfLycee}
\useproflyclib{piton}
\usepackage[executable=python]{pyluatex} % lua + shell-escape
```

### 2.2 Commande

```
\begin{ConsolePiton}[Options piton]<Clés>{Options tcbox}
...
...
\end{ConsolePiton}
```

Les clés, à placer entre <...>, sont :

- **<Logo>** pour afficher un petit logo dans les *titres* de la console REPL ; défaut : **<true>**
- **<Largeur>** pour spécifier la largeur de la console REPL ; défaut : **<\linewidth>**
- **<Alignement>** pour spécifier l'alignement de la console REPL. défaut : **<flush left>**

### 2.3 Exemples

```
%Déclaration d'une fonction python + librairie random pour utilisation ultérieure
\begin{python}
from random import randint

def valeur_absolue(x):
    "Renvoie la valeur absolue de x"
    #le petit test qui va bien
    if x > 0:
        return x
    else:
        return -x
\end{python}
```

```
\begin{ConsolePiton}{}
1+1
2**10
valeur_absolue(-3)
valeur_absolue(0)
valeur_absolue(5)
print(f"La valeur absolue de 5 est {valeur_absolue(5)}")
print(f"La valeur absolue de -4 est {valeur_absolue(-4)}")
\end{ConsolePiton}
```

◆ Début de la Console Python ◆

```
>>> 1+1
2
>>> 2**10
1024
>>> valeur_absolue(-3)
3
>>> valeur_absolue(0)
0
>>> valeur_absolue(5)
5
>>> print(f"La valeur absolue de 5 est {valeur_absolue(5)}")
La valeur absolue de 5 est 5
>>> print(f"La valeur absolue de -4 est {valeur_absolue(-4)}")
La valeur absolue de -4 est 4
```

◆ Fin de la Console Python ◆

```

\begin{ConsolePiton}<Largeur=11cm,Alignement=center,Logo=false>{}
1+1
2**10
valeur_absolue(-3)
valeur_absolue(0)
valeur_absolue(5)
print(f"La valeur absolue de 5 est {valeur_absolue(5)}")
print(f"La valeur absolue de -4 est {valeur_absolue(-4)}")
liste = [randint(1,20) for i in range(10)]
print(liste)
print(max(liste), min(liste), sum(liste))
\end{ConsolePiton}

```

— Début de la Console Python —

```

>>> 1+1
2
>>> 2**10
1024
>>> valeur_absolue(-3)
3
>>> valeur_absolue(0)
0
>>> valeur_absolue(5)
5
>>> print(f"La valeur absolue de 5 est {valeur_absolue(5)}")
La valeur absolue de 5 est 5
>>> print(f"La valeur absolue de -4 est {valeur_absolue(-4)}")
La valeur absolue de -4 est 4
>>> liste = [randint(1,20) for i in range(10)]
>>> print(liste)
[9, 4, 8, 3, 20, 16, 12, 2, 10, 2]
>>> print(max(liste), min(liste), sum(liste))
20 2 86

```

— Fin de la Console Python —

```

\begin{ConsolePiton}<Largeur=10cm,Alignement=center>{}
[i**2 for i in range(50)]
\end{ConsolePiton}

```

♣ Début de la Console Python ♣

```

>>> [i**2 for i in range(50)]
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196,
225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729,
784, 841, 900, 961, 1024, 1089, 1156, 1225, 1296, 1369, 1444,
1521, 1600, 1681, 1764, 1849, 1936, 2025, 2116, 2209, 2304,
2401]

```

♣ Fin de la Console Python ♣