

Développer avec Python

Langage interprété? Travailler en mode console ou programme? avec ou sans environnement?



Un algorithme est un objet théorique. Quand on veut l'exécuter sur une machine, il faut d'abord le transformer en **programme**. Le programme sera écrit dans un **langage de programmation**. Pour qu'il soit exécuté, il devra encore être traduit dans le **langage de la machine**. On n'écrit pas le programme dans le langage de la machine (trop fastidieux + langage différent selon le type de machine)

Exemples de langages de prog. : Python, Javascript, Java, C, PHP, ... Beaucoup d'autres.

Interpréteur Python = exécutable python.exe

Pour exécuter du Python, il nous faut un **exécutable** python.exe.

Même si Python est installé, il faut encore que Windows sache où le trouver!

Au lycée, on peut trouver une version de Python dans le répertoire C:\EduPython\App

Il est possible aussi de configurer Windows pour lui dire d'aller chercher automatiquement dans ce répertoire.

Pour la suite, on fera comme si ce réglage était fait.

Console sans environnement - Commande python en ligne de commande

Par exemple en ligne de commande cmd, j'appelle python

Windows ajoute automatiquement le .exe

```

C:\ Command Prompt
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

U:\ python
Python 3.7.0 (default, Jun 28 2018, 13:15:42)
[GCC 7.2.0] :: Anaconda, Inc. on Windows
Type "help", "copyright", "credits" or "licence" for more information.
>>> 3+2
5
>>>
  
```

Note : J'ai mis la console en noir sur blanc pour l'impression.

Dans la console, vous pouvez entrer des commandes, elles seront exécutées de suite.

Exemple : >>>3+2

Interprétation

>>>print("hello world !") est une ligne de programme en Python. Dès que l'on valide, l'interpréteur Python traduit cette commande dans un **langage compréhensible par la machine** pour qu'elle l'exécute.



Quand vous verrez >>> vous comprendrez que l'on est en mode console.

En mode console, quand une instruction produit un résultat, celui-ci est affiché.

Mode programme - fichier programme avec éditeur de texte

Plutôt que d'entrer un programme ligne après ligne dans la console, il est plus pratique de sauvegarder toutes les lignes du programme dans un fichier texte. C'est le **fichier programme**.

```

# fichier helloworld.py
print("hello world !")
  
```

On peut écrire ce fichier avec un éditeur de texte. On l'enregistre en lui donnant un nom se terminant par l'extension .py.

notepad++ est un très bon choix.



Ne surtout pas utiliser Word ou Wordpad ou Notepad pour éditer un fichier programme, ils ne sont pas du tout fait pour!



Grâce à l'extension, l'éditeur de texte détecte qu'il s'agit d'un fichier programme en Python et assure la **coloration syntaxique**, c'est à dire que les différents éléments reconnus sont colorés. L'écriture du programme s'en trouve facilitée.



Le fichier programme est parfois appelé **code** ou encore **script**. Le terme **script** est utilisé quand il s'agit d'un programme exécutant une tâche simple. *Exemple : supprimer tous les fichiers .tmp dans une arborescence.*

Exécution fichier programme, sans environnement

Par exemple, vous avez créé le fichier `helloworld.py` dans un certain répertoire. Dans un terminal Windows :

- se placer dans le répertoire,
- lancer `python helloworld.py`

L'interpréteur python lit le fichier ligne à ligne et le traduit dans le langage de la machine au fur et à mesure (*langage interprété*).



En mode programme, les résultats ne sont pas affichés à l'écran. Pour afficher quelque chose, il faut le demander explicitement, par exemple avec la commande `print`.

Utilisation d'un environnement

Un environnement est un logiciel qui inclut :

- un éditeur de texte,
- une console.

L'environnement n'apporte donc rien d'indispensable. Il facilite seulement le travail puisqu'il évite de passer de l'éditeur de texte à la ligne de commande Windows.

Les environnements préférés au Lycée sont EduPython, Pyzo, Spyder.



En tant que *spécialistes* de l'informatique, il me semble important que vous sachiez utiliser les deux méthodes. Néanmoins, nous utiliserons le plus souvent un environnement. Dans tous les cas, apprenez à ne pas être trop dépendant d'une solution. En informatique, il est important de savoir s'adapter.

Pour info : cas du Python dans internet

Il arrive parfois que des sites proposent une zones de texte permettant d'éditer un programme Python et ensuite de l'exécuter. Quel interprète est utilisé ?

- Le navigateur ne peut pas utiliser un interprète installé sur la machine elle-même, cela poserait trop de problèmes de sécurité.
- Le navigateur ne sait pas exécuter lui-même le programme Python. Certains sites contournent le problème en traduisant automatiquement le code Python en code Javascript. Le navigateur sait exécuter le Javascript.
- Sur les site plus sérieux (par exemple repl.it) le programme python est envoyé au serveur qui possède un interpréteur Python. Celui-ci exécute le programme et renvoie le résultat au navigateur.



Interprété / Compilé

- Python est **interprété** : à chaque exécution du programme, `python.exe` traduit au fur et à mesure, ligne après ligne, comme un interprète traduisant un discours en direct.
- D'autres langages sont **compilés** (C, C++) : On produit une traduction du fichier programme une bonne fois pour toutes, comme un traducteur qui traduit un livre.