

Les langages de programmation

- Les ordinateurs ne comprennent pas (ou pas encore) notre langage : il faut leur parler dans un langage prévu à cet effet.
- Il existe des centaines de langages (voir [Wikipédia](#) ou [ce site](#)).
- Les plus connus : C, C++, Java, Python, Javascript, PHP, ...

Comparaison de syntaxes

Voici un même programme, qui retourne un mot (exemple MOT → TOM), écrit dans 4 langages :

Pascal (1970)

```
const P : PChar = 'MOT';
var i:integer; len:integer;
begin
  len:= length(P);
  for i:=len-1 downto 0 do
    begin
      write(P[i]);
    end;
end;
```

Java (1995)

```
String s = new String("MOT");
for(int i = s.length()-1; i >=0 ; i--)
{
  char c = s.charAt(i);
  System.out.print(c);
}
```

PHP (1994)

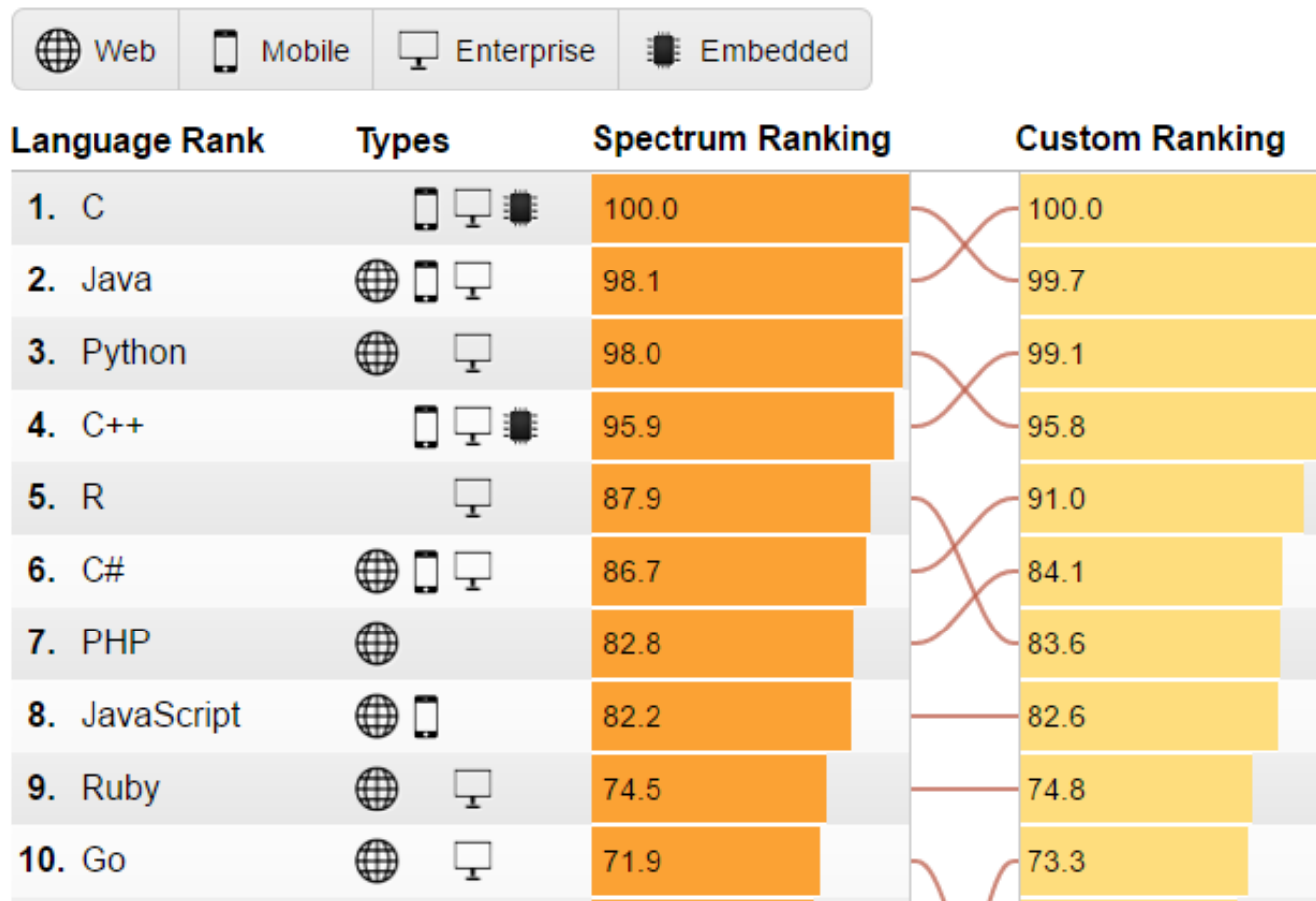
```
$str = "MOT";
$len = strlen($str);
for($i = $len-1; $i >= 0 ; $i--) {
  echo $str[$i];
}
```

Python (1990)

```
chaine = "MOT"
print(chaine[::-1])
```

Source : d'après <https://www.scriptol.fr/programmation/histoire-langages.php>

Top 10 des langages les plus populaires en 2016



Source :

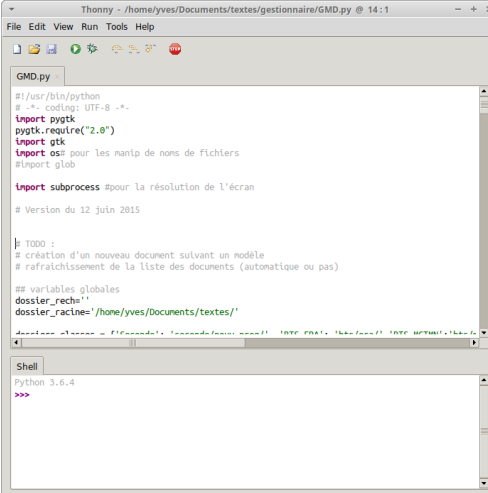
<https://www.developpez.com/actu/101905/Meilleurs-langages-de-programmation-en-2016-decouvrez-les-langages-en-forte-croissance-et-les-plus-demandes-par-les-employeurs-selon-l-IEEE/>

Installer Python

- Il y a plusieurs « distributions » de Python, nous allons utiliser Thonny.
- Si le logiciel est déjà installé sur votre poste, lancez-le ;
- sinon allez à l'adresse :

<https://thonny.org/>

téléchargez la version Windows, lancez-l'installation puis lancez Thonny.



```
GMD.py
#!/usr/bin/python
# -*- coding: UTF-8 -*-
import pygame
pygame.require("2.0")
import gtk
import os # pour les manip de noms de fichiers
import glob

import subprocess #pour la résolution de l'écran

# Version du 12 juin 2015

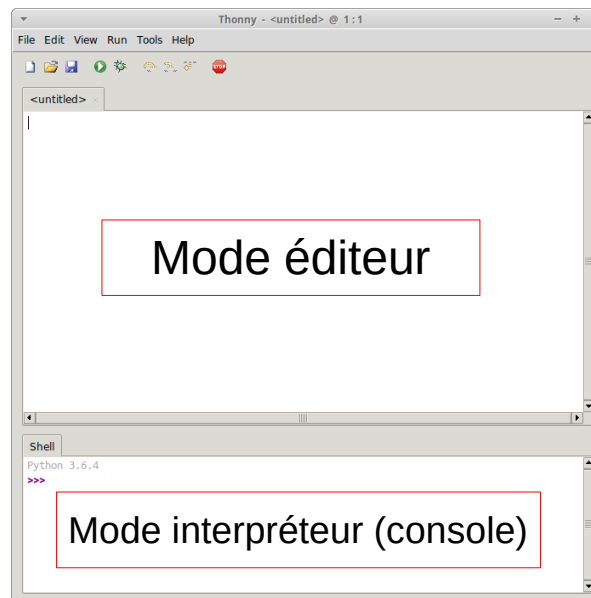
# TODO :
# création d'un nouveau document suivant un modèle
# rafraichissement de la liste des documents (automatique ou pas)

## variables globales
dossier_reche=""
dossier_ractne="/home/yves/Documents/textes/"

Shell
Python 3.6.4
>>>
```

Deux modes d'utilisation

- Il existe deux modalités d'utilisation de Python : le mode interpréteur (ou console) et le mode éditeur.
- Dans Thonny, les deux modes apparaissent dans la même fenêtre.



Mode interpréteur (console Python)

- Le symbole `>>>` vous invite à entrer des instructions ou des calculs.
- Tapez ceci (sans les `>>>`), et validez :

```
>>> Bonjour Python
```
- Python ne comprend visiblement pas le français (« `SyntaxError` » : erreur de syntaxe) ; ce qui est normal.

Mode interpréteur (console Python)

- Entrez les instructions suivantes (sans les >>>), en validant à chaque fois et en observant les réponses de Python :

```
>>> 5
```

```
>>> 7 - 3
```

```
>>> 1/3
```


```
>>> 2^3
```

```
>>> 2**3
```

```
>>> 3.14 + 1.05
```

```
>>> 3.14 + 2.05
```

- Remarques :

- Python comprend les mathématiques (le seul langage universel ) ;
- la virgule décimale est ici remplacée par un point (à l'anglo-saxonne) ;
- Python ne fait que du calcul arrondi avec parfois des résultats surprenants... ;
- la puissance se note ** et pas ^

Mode interpréteur (console Python)

- Poursuivons :

```
>>> valeur1 = 5.3
```

```
>>> valeur2 = 2.1
```

```
>>> valeur1
```

```
>>> valeur1*valeur2
```

```
>>> valeur1valeur2
```

- On a utilisé des **variables** (nommées ici « valeur1 » et « valeur2 ») pour stocker des valeurs.
- Le nom « valeur1valeur2 » est vu comme celui d'une variable, qui n'a pas été définie, l'interpréteur ne comprend pas ce qu'on lui demande...

Mode interpréteur (console Python)

- Parlons maintenant de texte à afficher ; tapez :

```
>>> bonjour
```

puis validez.

- Le nom « bonjour » est vu comme celui d'une variable (ou d'une fonction) inconnue, l'interpréteur ne comprend pas.
- Tapez maintenant :

```
>>> print("bonjour")
```
- Le mot **print** désigne une **fonction** de Python, qui sert à afficher un texte ou le contenu d'une variable. Les fonctions de Python 3 comportent systématiquement des parenthèses : `print(...)`, `input(...)`, etc.
- Remarque : une variable ne peut pas avoir le même nom qu'une fonction. Il existe une trentaine de noms de fonctions réservés en Python : voir sur [Wikibooks](#).

Mode interpréteur (console Python)

Pour comprendre la différence entre du texte et des noms de variables ; tapez ceci :

```
>>> bonjour=10
>>> print(bonjour)
>>> print("bonjour")
>>> bonsoir="bonjour"
>>> print(bonsoir)
```

Mode interpréteur (console Python)

- Une variable peut contenir du texte :

```
>>> a = "bonjour"
```

```
>>> print("a")
```

```
>>> print(a)
```

```
>>> b = "tout le monde"
```

```
>>> c = a+b
```

```
>>> print(c)
```

- Ce type de variable s'appelle une **chaîne de caractères** (on parlera aussi de *texte*).
- Si on ajoute deux chaînes, on obtient une nouvelle chaîne.

Mode interpréteur (console Python)

- Tapez maintenant ceci :

```
>>> a = 3
```

```
>>> print(2*"a")
```

```
>>> print(2*a)
```

- Attention de ne pas confondre le texte « a » et la variable nommée a.

Mode interpréteur (console Python)

- On peut mélanger du texte et des nombres :
 - >>> print("1+1 font ",1+1)
 - >>> a = 7
 - >>> print("Le carré de ",a," est ",a**2)
- La virgule sert à séparer les éléments à afficher.

Mode interpréteur (console Python)

- Le nom des variables n'a pas d'importance, sauf pour la lisibilité du programme. Ainsi les deux programmes suivants font la même chose :

```
>>> a = 250
>>> b = 10
>>> print(a+b*a/100)
```


```
>>> ancien_prix = 250
>>> pourcentage = 10
>>> print("Le nouveau prix est ",ancien_prix+pourcentage*ancien_prix/100)
```

- Mais le second est plus compréhensible...

Mode interpréteur (console Python)

- L'interpréteur est pratique pour tester des instructions.
- Le problème est que quand vous quittez l'interpréteur, toutes vos instructions sont perdues...

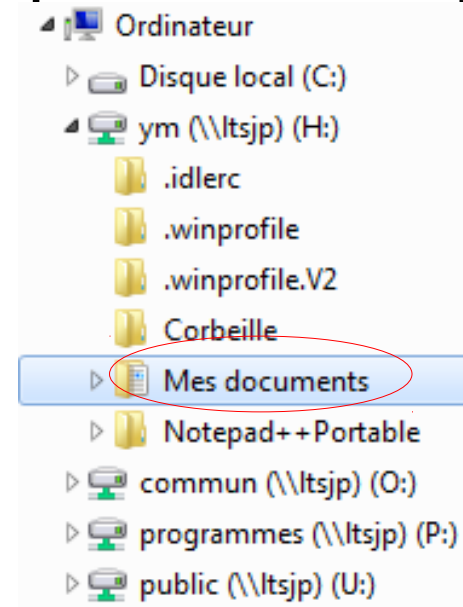
Mode éditeur

- Dans le mode éditeur, on va pouvoir écrire des programmes longs puis ensuite l'exécuter avec le bouton  (ou F5).

Le résultat s'affichera dans la console.

- Pensez à sauvegarder vos programmes de sorte qu'il ne soient pas perdus quand vous quittez Thonny...

- **Attention** : il faudra sauvegarder vos documents dans votre dossier réseau H :



Mode éditeur

- Il y a des différences de fonctionnement avec le mode interpréteur.

- Par exemple, tapez dans l'éditeur :

`3+5`

puis lancez l'exécution avec le bouton 

- Contrairement au mode interpréteur, le calcul `3+5` ne se fait pas et rien ne s'affiche.

- Par contre,

`print(3+5)`

donnera le résultat attendu.

Ressources

- Tutoriel vidéo
- Pythonneries (une trentaine de vidéos)
- Débuter avec Python au lycée.
- Un très bon cours sur le site openclassrooms (ex Site du zéro)
- Apprendre à programmer avec Python 3

Des langages illisibles !

Voici un programme, qui affiche « Hello world ! » dans deux langages exotiques :

Brainfuck	Ook (pour les orang-outans !)
<pre>>+[[>],.----- ----- --- [+++++ +++++ ++ + [<]]> << [<]> [-]> [>]< [.<]</pre>	<pre>Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook! Ook! Ook? Ook! Ook. Ook? Ook. Ook. Ook. Ook! Ook. Ook. Ook? Ook. Ook. Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook! Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook. Ook? Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook? Ook. Ook! Ook. Ook? Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook. Ook. Ook? Ook. Ook! Ook. Ook. Ook? Ook! Ook. Ook? Ook?</pre>

Source : <https://fr.wikipedia.org/wiki/Brainfuck>