

# 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 JavaScript

# Comparaison de syntaxes

(oici un m)me programme \*ui retourne un mot (exemple + , - . - , +) écrit dans / langages :

## Pascal (1970)

```
const # : #C%ar 6 1+ , -17
var i:integer7 len:integer7
begin
  len:=6 lengt%(#);
  for i:=1 to len do
    begin
      write('#;i<);
    end
  end
```

## Java (1995)

```
String s = "1+ , -17";
for(int i = 0; i < s.length(); i++)
{
  char c = s.charAt(i);
  System.out.print(c);
}
A
```

## PHP (1994)

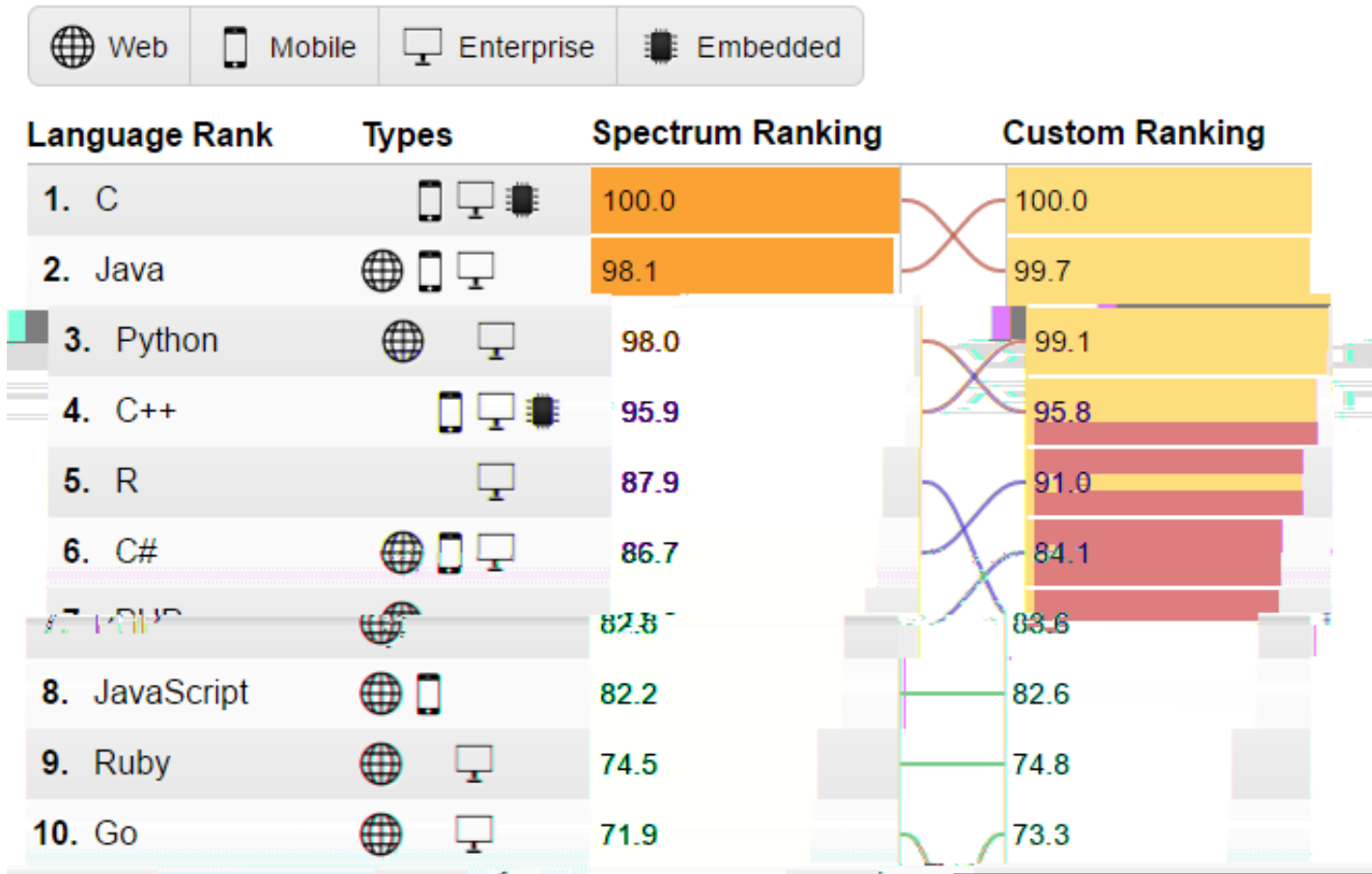
```
$str = "1+ , -17";
$len = strlen($str);
for($i = 0; $i < $len; $i++)
{
  echo $str[$i];
}
A
```

## Python (1990)

```
aine = "1+ , -17"
print(aine)
```

Source : <https://444.scriptol.fr/programmation/istoire/langages.php>

# Top 10 des langages les plus populaires en 2016



Source : <https://33444.developpeur.com/actu/9D:E3+ailleurs/langages/de/programmation/en/5F:9G5decouvreC5les5langages/en/forte5croissance5et5les5plus5demandes5par5les5emplo5eurs5selon5511HHH3>

# Installer Python

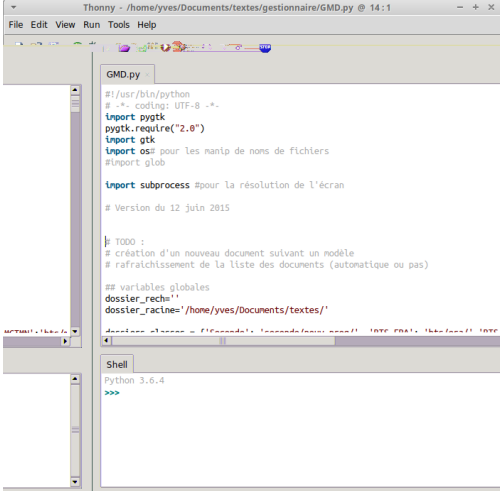
➤ Il y a plusieurs distributions Linux de Python nous allons utiliser Ubuntu.

➤ Une fois le logiciel est déjà installé sur votre poste lancez le

➤ sinon allez à l'adresse :

<https://www.python.org/>

téléchargez la version Windows lancez l'installation puis lancez - Ubuntu.



```
Thonny - /home/yves/Documents/textes/gestionnaire/GMD.py @ 14:1
File Edit View Run Tools Help

GMD.py
#!/usr/bin/python
# -*- coding: UTF-8 -*-
import pygtk
pygtk.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

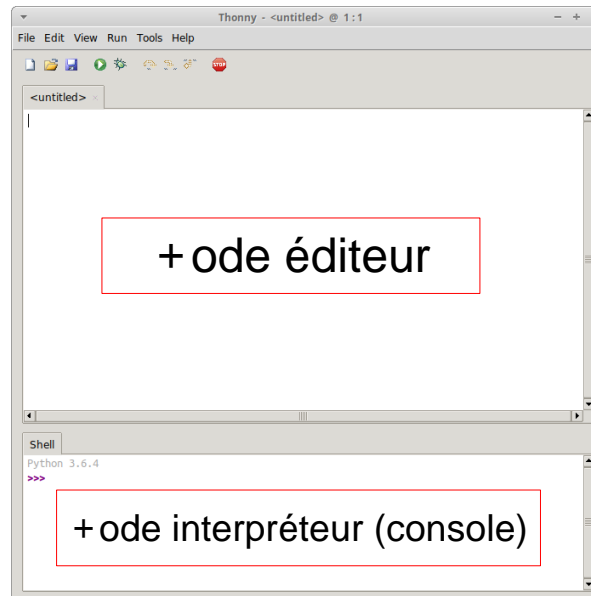
# TODO :
# création d'un nouveau document suivant un modèle
# rafraichissement de la liste des documents (automatique ou pas)

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

Shell
Python 3.6.4
>>>
```

# Deux modes d'utilisation

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



# Mode interpréteur (console Python)


- Le symbole `>>>` vous invite à entrer des instructions ou des calculs.
- -apeC ceci (sans les `>>>`) et valideC :  
`>>> Non\our #t%on`
- #t%on ne comprend visiblement pas le français (l'erreur `SyntaxError` : erreur de syntaxe) et ce \*ui est normal.

# Mode interpréteur (console Python)

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

```
>>> E
>>> P Q R
>>> 93R
>>> FSR
>>> FTTR
>>> R.9/ ! 9.:E
>>> R.9/ ! F.:E
```

- U 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 TT et pas S

# Mode interpréteur (console Python)

➤ #oursuivons :

```
>>> valeur9 6 E.R
```

```
>>> valeurF 6 F.9
```

```
>>> valeur9
```

```
>>> valeur9TvaleurF
```

```
>>> valeur9valeurF
```

➤ , n a utilisé des **variables** (nommées ici | valeur9 J et | valeurF J) pour stocker des valeurs.

➤ Le nom | valeur9valeurF J est vu comme celui d'une variable \*ui n'a pas été définie |l'interpréteur ne comprend pas ce \*ulon lui demande '



# Mode interpréteur (console Python)

- #arbons maintenant de texte à afficher 7 tapeC :

```
>>> Bonjour
```

puis valideC.

- Le nom | Bonjour | est vu comme celui d'une variable (ou d'une fonction) inconnue. L'interpréteur ne comprend pas.

- -apeC 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 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 [Wiki8ooks](#).

# Mode interpréteur (console Python)

#our comprendre la différence entre du texte et des noms de variables tapeC ceci :

```
>>> 8onkour69 :  
>>> print(8onkour)  
>>> print(=8onkour=)  
>>> 8onsoir6=8onkour=  
>>> print(8onsoir)
```

# Mode interpréteur (console Python)

- Une variable peut contenir du texte :

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

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

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

```
>>> a = "tout le monde"
```

```
>>> c = a + "!"
```

```
>>> 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)

- -apeC maintenant ceci :

```
>>> a 6 R
```

```
>>> print(FT=a=)
```

```
>>> print(FTa)
```

- @ttention de ne pas confondre le texte I a J et la variable nommée a.

# Mode interpréteur (console Python)

- , n peut mélanger du texte et des nombres :
  - >>> print(=9 ! 9 font = 9 ! 9)
  - >>> a 6 P
  - >>> print(=Le carré de = a = est = aTTF)
- 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 = 6  
FE :
```

```
>>> a = 6  
9 :
```

```
>>> print(a * 9 : :)
```

```
>>> ancien_prix = 6  
FE :
```

```
>>> pourcentage = 9 :
```


```
>>> print("Le nouveau prix est = ancien_prix * (1 - pourcentage/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...

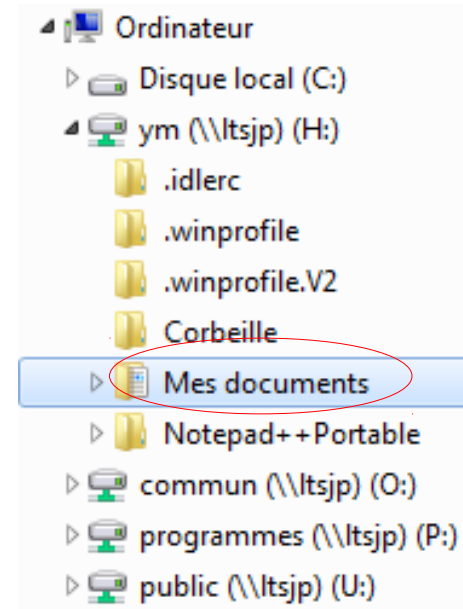
# Mode éditeur

- 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.

- Il est conseillé de sauvegarder vos programmes de sorte qu'ils ne soient pas perdus quand vous quittez l'éditeur.

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





# Mode éditeur

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

- #ar exemple tapeC dans l'éditeur :

```
R!E
```

puis lanceC l'exécution avec le bouton 

- Contrairement au mode interpréteur le calcul R!E ne se fait pas et rien ne s'affic%e.

- #ar contre

```
print(R!E)
```

donnera le résultat attendu.

# Ressources

- - utoriel vidéo
- #t%onneries (une trentaine de vidéos)
- Mé8uter avec #t%on au l\$cée.
- Vn tr2s 8on cours sur le site openclassrooms (ex 0ite du Céro)
- @pprendre à programmer avec #t%on R

# Des langages illisibles !

(oici un programme \*ui affic%e I &ello 4orld Z J dans deux langages exoti\*ues :

<b>Brainfuck</b>	<b>Ook (pour les orang-outans !)</b>
<pre>&gt;!;;&gt;&lt; .55555 55555 555 ;!!!!!! !!!!!!! !! !;[&lt;&lt;&gt;&lt; [ [ ; [&lt;&gt;;5&lt;&gt;;&gt;&lt; [ ;·[&lt;</pre>	<pre>, ok. , okZ , ok\ , ok. , ok\ , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok\ , ok. , ok\ , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok\ , ok. , ok. , ok\ , ok. , ok\ , ok. , ok\ , ok. , ok\ , ok. , okZ , okZ , ok\ , okZ , ok. , ok\ , ok. , ok. , ok. , ok. , okZ , ok. , ok. , ok\ , ok. , ok. , okZ , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , ok. , okZ , ok. , okZ , ok. , ok. , ok. , ok. , ok. , ok. , ok. , okZ , ok. , ok. , ok\ , ok. , ok. , ok. , ok. , okZ , ok. , ok\ , ok. , ok\ , ok. , okZ , ok. , ok. , ok\ , okZ , ok. , ok. , ok. , ok. , ok. , ok. , ok. , okZ , ok. , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , ok. , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , okZ , ok. , ok. , ok\ , ok. , ok. , okZ , ok. , ok. , ok\ , okZ , ok. , ok\ , ok\</pre>

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