

Apprentissage de Python – Fiche 01

1. Introduction

En 1989, le hollandais **Guido van Rossum** commence le développement du langage de programmation Python.

Python est un langage **multiplateforme**, c'est-à-dire disponible sur plusieurs architectures (compatible PC, tablettes, smartphones, ordinateur low cost Raspberry Pi...) et systèmes d'exploitation (Windows, Linux, Mac, Android...).

Le langage Python est gratuit, sous **licence libre**.

C'est un des langages informatiques les plus populaires avec C, C++, C#, Objective-C, Java, PHP, JavaScript, Delphi, Visual Basic, Ruby et Perl ([liste non exhaustive](#)).

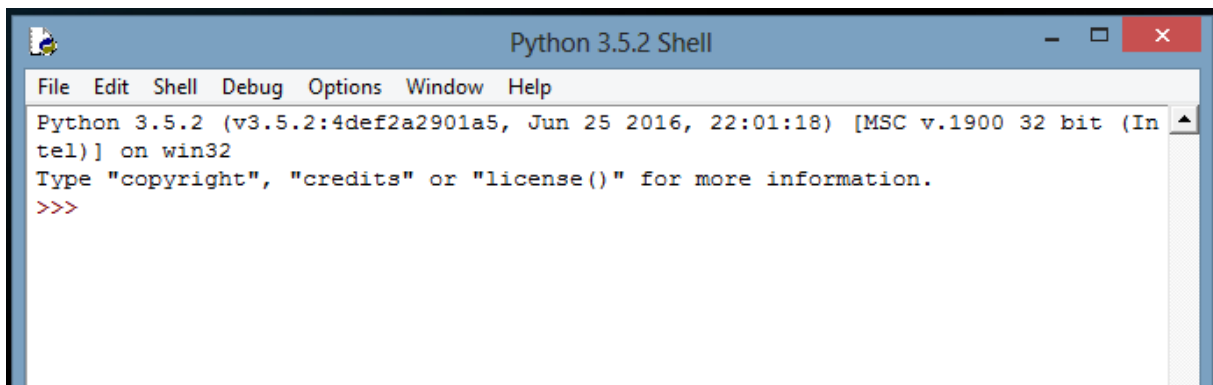
Actuellement, Python en est à sa version 3.

Cependant, la version 2 est encore majoritairement utilisée.

Attention : Python 2 n'est pas compatible avec Python 3 !

Python peut s'utiliser de deux façons :

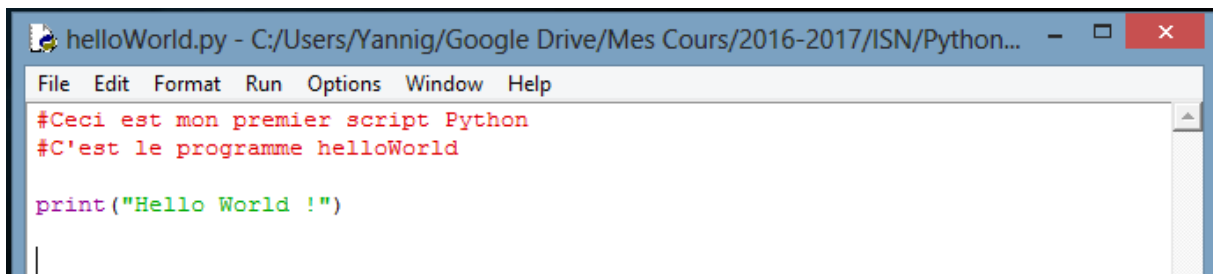
- Directement dans l'interpréteur :



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

Dans ce cas, on entre les instructions après les 3 signes supérieur (>>>) et lorsqu'on appuie sur entrée Python exécute la ligne de code.

- Dans un script :



```
helloWorld.py - C:/Users/Yannig/Google Drive/Mes Cours/2016-2017/ISN/Python...
File Edit Format Run Options Window Help
#Ceci est mon premier script Python
#C'est le programme helloWorld

print("Hello World !")
|
```

Un programme est une séquence d'instructions.

Dans le cas d'un programme en langage Python, on parle souvent de **script Python**.

Un script se présente sous la forme d'un fichier texte avec l'extension **.py**

Dans un premier temps nous travaillerons directement dans l'interpréteur, nous ne pourrons donc pas sauvegarder nos codes. Très rapidement nous écrirons nos scripts dans un fichier avec l'extension `.py` pour pouvoir les enregistrer.

Le script Python peut être produit avec n'importe quel traitement de texte, mais Python contient normalement un programme (IDLE) à cet effet.

2. Les variables

En programmation nous avons besoin de stocker des données, et pour cela nous utilisons des variables.

Le nom d'une variable s'écrit avec des lettres (non accentuées), des chiffres ou bien l'underscore `_`.
Le nom d'une variable ne doit pas commencer par un chiffre.

En langage Python, l'usage est de ne pas utiliser de lettres majuscules pour nommer les variables (celles-ci sont réservées pour nommer les classes).

Ces variables peuvent être de différents types (`int`, `float`, `Str`, ...)

Python utilise le système de typage dynamique, c'est-à-dire que nous n'avons pas besoin de définir les variables, elles sont automatiquement définies lorsqu'elles apparaissent dans le programme et Python leur attribue le type le plus proche de la valeur qu'on leur affecte.

Exemple : dans l'interpréteur, saisir les lignes suivantes et observer les résultats obtenus :

```
>>> a=2
```

La fonction `print()` affiche la valeur de la variable :

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

La fonction `type()` affiche le type de la variable `a` :

```
>>>type(a)
```

```
>>>b=3.1
```

```
>>>type(b)
```

```
>>>c= « hello »
```

```
>>>type(c)
```

3. Calculer avec Python

Opérateurs mathématiques :

- `*` multiplication
- `-` soustraction
- `/` division (`>>>20/3` affiche :)
- `//` division entière (`>>>20//3` affiche :)
- `+` addition
- `**` puissance (`>>>2**3` affiche :)
- `%` reste de la division entière (`>>>13%2` affiche : ; `>>>12%2` affiche :)

Python respecte les priorités opératoires : (>>>3+2*5 affiche :)

Remarque : Le symbole + effectue la concaténation si les variables sont du type String

Exemple :

```
>>>a= « Bonjour »
>>>b= « tout le monde »
>>>c= a + b
>>>print(c)  affiche : .....
```

4. Interactions avec l'utilisateur

Nous avons régulièrement besoin que l'utilisateur du programme rentre des valeurs au clavier. Pour cela nous utiliserons la fonction input(). On peut rajouter des arguments à cette fonction dans la parenthèse. Saisir les exemples suivants :

```
>>>nom=input(« Entrez votre nom »)
>>>print(nom)      affiche : .....
```

La fonction print() permet d'afficher les contenus de plusieurs variables en un seul appel, séparés par un espace :

```
>>>a= « Bonjour »
>>>b= « les élèves d'ISN »
>>>print(a,b)      affiche : .....
```

5. Les conditions (if else)

Dans la pratique, nous devons régulièrement tester les valeurs contenues dans des variables pour savoir quelles instructions effectuer. Pour utiliser la structure if else, on écrit notre script dans un fichier en .py. La syntaxe est la suivante :

If condition :

Instructions 1

else :

Instructions 2

Copier les lignes de code ci-dessous dans un fichier texte, puis enregistrez-le sous le nom `essai1.py` dans votre espace personnel :

Remarque : Les lignes qui commencent par # sont des commentaires sont ignorées lors de l'exécution du script. Elles sont nécessaires pour la clarté du script, il faut prendre l'habitude de commenter vos programmes.

```
# script essai1.py

chaine = input("Note sur 20 : ")
note = float(chaine)          #on convertit la variable chaine dans le type float
if note >= 10.0:
    # ce bloc est exécuté si l'expression (note >= 10.0) est vraie
    print("J'ai la moyenne")
else:
    # ce bloc est exécuté si l'expression (note >= 10.0) est fausse
    print("C'est en dessous de la moyenne")
print("Fin du programme")

#Cette ligne est nécessaire pour quitter le programme
input(« Appuyez sur la touche Entrée »)
```

Pour exécuter le programme il suffit de double-cliquer sur le fichier essai1.py

6. Vidéos

En complément de ce document, vous pouvez visionner les pythonneries N°1, 2 et 3 sur Dailymotion :

http://www.dailymotion.com/playlist/x22t3u_universal_avs_apprendre-le-python/1#video=xbk6lj

7. Exercices

1. Saisir les instructions suivantes et observer les résultats :

```
>>>a= « Elorn »
>>>b=len(a)
>>>print(b)                Affichage : .....
Que fait la fonction len() ? .....
>>>print(a[1])             Affichage : .....
>>>print(a[4])             Affichage : .....
>>>print(a[5])             Affichage : .....
Comment afficher la première lettre de la variable a : .....
```

2. Ecrire une séquence d'instructions qui demande à l'utilisateur son nom puis affiche la phrase : « Bonjour *nom* , bienvenue en ISN » où *nom* sera remplacé par le nom de l'utilisateur.

3. Saisir les instructions suivantes et observer les résultats :

```
>>>taille=input( « Entrez votre taille en cm »)
```

```
>>>print(taille)
```

```
>>>taille2=taille+10
```

Que se passe-t-il ?

```
>>>type(taille)           Affiche : .....
```

```
>>>type(10)              Affiche : .....
```

La fonction input retourne une variable de type et 10 est du type, or Python ne sait pas faire une addition entre deux variables de types différents.

Pour résoudre ce problème, nous allons utiliser la fonction int() qui convertit une variable de type String et type int :

```
>>>taille=int(taille)
```

```
>>>type(taille)           Affiche : .....
```

4. Ecrire une séquence d'instruction qui demande à l'utilisateur son nom, son âge, puis affiche la phrase suivante : « Bonjour *nom*, en 2030 vous aurez *age* ans » où *nom* et *age* seront remplacés par les vraies valeurs.

5. Ecrire un script qui simule le jeu du juste prix :

L'ordinateur génère un nombre aléatoire compris entre 0 et 100 (l'instruction pour cela est : `nbAleat=random.randint(0,100)` et il faut rajouter au début du script la ligne : `import random`).

L'utilisateur essaie de retrouver ce nombre, il fait des propositions et l'ordinateur lui répond « C'est plus ! » ou « C'est moins ! » ou « C'est gagné ».

En option vous pouvez rajouter un compteur et afficher à la fin une phrase du type : Bravo, vous avez gagné en coups !

6. Revenons à notre mini-projet .

Ecrire un script qui demande à l'utilisateur de saisir un nombre en binaire, puis le convertit en écriture décimale. S'il vous manque des instructions, demandez au professeur ou faites des recherches sur internet.

7. Ecrire un script qui demande à l'utilisateur de saisir un nombre en décimale, puis le convertit en écriture binaire. Dans un premier temps vous pourrez afficher le nombre en binaire « à l'envers », écriture de droite à gauche, puis vous l'écrirez dans le bon sens.

8. Regroupez les scripts 6 et 7 dans un programme plus complet.

9. Rajouter les conversions dans une base quelconque.