

J'apprends à programmer avec Python

Atelier d'initiation à la programmation





Zokora Elvis

degbagnon@gmail.com

@ElvisZokora

- ING Génie Logiciel
- Consultant Sys et Sécurité
- Amoureux du langage Python
- Président de Python CI
- Vice –président Comité Django Girls Abidjan

Présentation du langage Python

Prérequis :

Disposer d'un ordinateur

Une grande envie d'apprendre à coder passion pour le codage informatique

La curiosité

La patience

La rigueur

Objectifs

- Connaitre Python et savoir ce qu'il fait
- Mettre le pied à l'étrier : Faire vos premiers pas en programmation Python (version 3)
- Mettre l'accent sur la pratique
- Donner les clés pour:
 - Approfondir rapidement vos connaissances en toute autonomie
 - Concevoir rapidement de vraies applications utiles

Contenu de la formation

1. Variable
2. Type de variable
3. Fonctions de bases
4. Opération de base
5. Conditions et boucles
6. Listes et Tuples
7. Dictionnaires et Fonctions
8. Modules
9. Fichiers
10. La POO
11. Introduction aux interfaces graphiques
12. Introduction à la programmation Web
13. Introduction à la programmation Mobile

Qu'est-ce qu'un programme?

Définition : Un programme est une suite d'instructions/d'ordres compréhensibles par un ordinateur.

Objectif : Le programme va nous permettre de réaliser des applications / tâches

Exemple de programme : Word , Excel , DropBox , Firefox , Chrome ,

Le langage de programmation permet le dialogue entre le programmeur et l'ordinateur

Exemple de langage : C , C++ , Perl , Java , COBOL , Ruby , Php , Python , Html5 , Javascript , CSS , Xml ,

Le langage que qu'on va apprendre aujourd'hui est : Python

Python c'est quoi ?

- Un langage de programmation interprété
Ecrit par programmeur Guido van Rossum développeur Néerlandais
<https://gvanrossum.github.io/>
Fan de la série télévisée Monty Python's Flying Circus, il décide de baptiser ce projet Python.
- Multi-plateformes
- Open source
- Très utilisé dans l'industrie et le monde académique

- Orienté objet

Les objets sont des structures de données complexes

- De haut niveau

Gestion automatique de la mémoire

Système de gestion d'exceptions

...

- À typage dynamique

Le programmeur ne définit pas explicitement le type des variables

Python pour quoi ?

- Un langage facile à apprendre !
- Une syntaxe légère, efficace et agréable à lire
Une syntaxe discrète, code court et facile à lire
- Un système de délimitation des blocs de code basé sur l'indentation
Singulière, originale mais surtout efficace!

- Un cycle de développement (très) court
- Un langage multi-plateforme

Permet d'écrire des logiciels portables:

Pas besoin de créer une version spécifique de votre programme pour chaque système d'exploitation

Il marchera sans modification (à quelques rares exceptions près) sur:

Windows

MacOS

Linux

De nombreuses autres plateformes

- Une énorme base de développeurs, d'utilisateurs et de bibliothèques externes

Un langage très généraliste

Bibliothèque standard + les nombreuses bibliothèques externes = un potentiel énorme!

On peut presque tout faire avec Python

- interfaces graphiques
- calcul scientifique (algèbre, analyse, statistiques/probas, théorie des graphs, visualisation, etc.)
- calcul formel: calculer la dérivée ou l'intégrale d'une fonction, résoudre une équation, factorisation, calcul de limites, résolution de systèmes d'équations, ...

- informatique haute performance / calcul massivement parallèle
- chimie, biologie, astronomie, psychologie, finance, etc.
- optimisation, apprentissage automatique, traitement du signal, ...
- serveurs
- multimédia
- 3D
- Jeux
- traitement d'image
- applications web
- systèmes embarqués (robotique, raspberrypi, ...)
- bases de données
- réseau

- outils système
- greffons (plugins) pour Blender, The Gimp, Inkscape, FreeCAD, OpenOffice, etc.
- manipulation/édition de fichiers, documents, ... (PDF, HTML, XML, etc.)
- cartographie
- ...

Python est lisible

```
class CompteBancaire(object):  
    "Gestion d'un compte bancaire"  
  
    #Constructeur de la classe  
    def __init__(self,nom="Dupont",solde=1000):  
        self.nom=nom  
        self.solde=solde  
  
    #Methode permettant de gerer les depots  
    def depot(self,montant):  
        self.solde=self.solde+montant  
  
    #Methode permettant de gerer les retrait sur le compte Bancaire  
    def retrait(self,montant):  
        self.solde=self.solde-montant  
  
    #Methode permettant d'afficher le nom du Titulaire du compte et le solde du compte  
    def affiche(self):  
        print("le solde du compte bancaire de ",self.nom,"est de ",self.solde)
```

Exemple de code

Python:

```
l = [2, 3, 5]
for n in l:
    print(n)
```

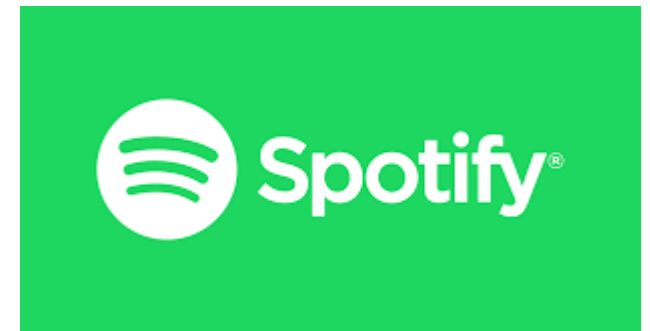
L'équivalent en C++ :

```
#include <vector>
#include <iostream>
int main() {
    std::vector<int> v {2, 3, 5}; // Ne
    marche pas pour C++98
    std::vector<int>::iterator it;
    for(it = v.begin() ; it != v.end() ;
    it++) {
        std::cout << *it << std::endl;
    }
    return 0;
}
```

L'équivalent en Java:

```
import java.util.ArrayList;
import java.util.Arrays;
public class Test {
    public static void main(String[]
    args) {
        ArrayList<Integer> list =
        new
        ArrayList<Integer>(Arrays.asList(
        2, 3, 5));
        for(int val : list) {
            System.out.println(val);
        }
    }
}
```

il n'y a pas que les petits qui l'utilisent



Installation des Outils et présentation de l'environnement de programmation

Installation de Python : la méthode classique...

www.python.org

L'environnement de développement IDLE

IDLE

- L'IDE "officiel" de Python
- Intégré de base avec Python (standard): rien à installer
- Écrit en Python
- Open source

L'environnement de développement IDLE

Pourquoi IDLE ?

- Un classique
- Intégré de base avec Python: si vous avez Python, vous avez IDLE (sauf cas particuliers)
- Fonctionne sur Windows, MacOSX, Linux, ...
- Fonctionnalités basiques mais adaptées aux besoins de cet atelier
- Ni trop simple, ni trop compliqué
- L'interprète python
- Ouvrir l'interprète en mode interactif

Les alternatives à IDLE

Si IDLE ne vous plaît pas

PyCharm

Spyder

Et les autres: Notepad++, Code::Blocks, Eclipse, ...

Écriture de programmes

Python avec IDLE

1. Démarrer IDLE

Taper idle depuis une console

2. Créer un nouveau programme

"File / New File" (Ctrl + n)

Une nouvelle fenêtre apparaît :
l'éditeur de code Python

3. Écrire notre programme

Écrire le programme suivant dans la fenêtre d'édition

```
print("hello")
```

4. Sauvegarder notre programme dans un fichier Python (.py)

"File / Save" (Ctrl + s)

5. Exécuter notre programme

"Run / Run module" (F5)

Le résultat est affiché dans la console IDLE ("Shell")

Ouvrir un programme existant

"File / Open" (Ctrl + o)

Utilisation de l'interprète interactif dans IDLE

1. Revenez sur la console (ou le fenêtre "Shell")

Ou depuis le menu: "Run / Python Shell"

2. Tapez directement la ligne suivante:

```
print("hello")
```

Astuce : pour copier coller plus vite dans la console IDLE, sélectionnez le texte et appuyez sur la touche Entrer

Utilisation de l'interpréteur Python depuis un terminal

Lancer l'interprète Python depuis un terminal

Windows: `python`

MacOSX, Linux, Unix: `python3`

Taper la ligne suivante dans l'interprète

```
print("hello")
```

Taper `exit()` pour quitter l'interprète

Calculer avec Python

Les trois caractères « supérieur à » constituent le signal d'invite, ou prompt principal, lequel vous indique que Python est prêt à exécuter une commande.

Par exemple, vous pouvez tout de suite utiliser l'interpréteur comme une simple calculatrice de bureau. Veuillez donc vous-même tester les commandes ci-dessous (Prenez l'habitude d'utiliser votre cahier d'exercices pour noter les résultats qui apparaissent à l'écran) :

```
>>> 5+3
```

```
>>> 2-3
```

```
>>> (5+3)*10
```


>>> 20/3

>>> 20//3

Variables et types de base

Qu'est-ce qu'une variable ?

C'est un espace de stockage dans la mémoire de l'ordinateur (une "case") caractérisé par:

- Un nom: qu'on utilise pour désigner la variable dans le programme
Exemples: "var", "x", "toto", ...
- Une valeur: le contenu de la variable
- Un type: on verra ça juste après
- Une portée: on verra ça plus tard...

Un programme utilise des variables pour stocker des données temporaires

Règles de nommage

Sous Python, les noms de variables doivent en outre obéir à quelques règles simples :

- Un nom de variable est une séquence de lettres ($a \rightarrow z$, $A \rightarrow Z$) et de chiffres ($0 \rightarrow 9$), qui doit toujours commencer par une lettre.
- Seules les lettres ordinaires sont autorisées. Les lettres accentuées, les cédilles, les espaces, les caractères spéciaux tels que \$, #, @, etc. sont interdits, à l'exception du caractère _ (souligné).
- La casse est significative (les caractères majuscules et minuscules sont distingué

il faut encore ajouter que vous ne pouvez pas utiliser comme nom de variables les 33 « mots réservés » ci-dessous (ils sont utilisés par le langage lui-même) :

```
and as assert break class continue def
del elif else except False finally for
from global if import in is lambda
None nonlocal not or pass raise return
True try while with yield
```

Exemples de noms corrects:

$x = 3$

$AbC = 3$

$x8 = 3$

$une_variable = 3$

Exemples de noms incorrects:

$une\ variable = 7$

$supervariable! = 19$

$8x = 19$

$une-variable = 3$

Conventions de nommage

En plus des règles précédentes, il existe des conventions (i.e. pas obligatoires mais recommandées)

Pour les noms de variables:

- éviter d'utiliser des lettres en majuscule (les noms avec des majuscules sont réservés à autre chose)
- utiliser le caractère underscore (_) pour les noms composés

Exemple:

```
nom_etudiant = "John"
```

```
nom_prof = "Alice"
```

Qu'est ce qu'une affectation ?

une affectation, aussi appelée assignation par anglicisme, est une structure qui permet d'attribuer une valeur à une variable.

En Python comme dans de nombreux autres langages, l'opération d'affectation est représentée par le signe égale =

Exemple:

```
x = 3
```

```
y = 18
```

- x est le nom de la 1ère variable et 3 sa valeur
- y est le nom de la 2e variable et 18 sa valeur

Merci !

Questions?
