

langage python

Table des matières

I - Présentation du langage Python.....	3
1. Langage python.....	3
II - Prise en main de l'interpréteur Python.....	5
1. Installation et première utilisation de python.....	5
III - Syntaxes élémentaires , types , boucles.....	9
1. Titre_Section1 du chapitre englobant.....	9
IV - fonctions python.....	10
1. Titre_Section1 du chapitre englobant.....	10
V - Annexe – Bibliographie, Liens WEB + TP.....	12

1. Bibliographie et liens vers sites "internet"	12
2. TP.....	12

I - Présentation du langage Python

1. Langage python

Python est un **langage** informatique **interprété** à usage généraliste , simple à apprendre et à utiliser, qui est essentiellement utilisé pour :

- **coder des petits scripts ou programmes simples**
- **piloter/orchestrer des appels vers des fonctions prédéfinies efficaces** (quelquefois codées en langage "C" de bas niveau et rapide)
- **piloter/orchestrer des calculs scientifiques**

Le langage python peut également être utilisé dans d'autres domaines (contrôles d'affichages graphiques, accès à des bases de données ou des fichiers, sites web, ...) sans cependant se démarquer d'autres langages informatiques (également bien adaptés pour effectuer efficacement ces tâches) .

1.1. Principales caractéristiques du langage python

simple	code facilement compréhensible et apprentissage rapide
interprété (et pas compilé)	comme les langages "basic" , "javascript" , "perl" , "ruby" , <i>avantages : souplesse et résultats immédiats</i> <i>inconvénients : exécution pas rapide (si 100% python)</i>
à usage général	on trouve des bibliothèques de fonctions prédéfinies pour presque tous les usages (calculs , accès bases de données , affichages , ...)
multi-paradigme	code procédural et séquentiel pour scripts simples code en mode "orienté objet" pour applications élaborées ...
mature et bien implanté	python existe depuis plus de 25 ans et est beaucoup utilisé dans le domaine scientifique et dans le cadre de l'administration système (ex : linux, ...)
open-source	accès et usage libre et gratuit . Seules quelques extensions "clefs en main" sont quelquefois payantes .
multi-plateformes	étant interprété, le langage python peut facilement être utilisé sur tout type de plateformes (Windows , Linux , Mac , smartphones, ...)
syntaxe avec indentations	contrairement à beaucoup d'autres langages qui délimitent des blocs de code via { et } ou via "begin" et "end" , le langage python n'utilise pas de délimiteur mais a une structure de code contrôlée par des indentations (décalages par rapport de débuts des lignes)

Bien qu'intéressant sur bien des points, le langage Python n'est pas interprété par les navigateurs "web/internet" . Les navigateurs "IE , Firefox , Chrome, ..." ont historiquement fait le choix d'utiliser le langage interprété javascript (bien aussi et un peu plus rapide que python) .

1.2. Historique et évolution

Première version publique officielle : 0.9.0 en **1991** développé par "[Guido van Rossum](#)" (Pays bas , Amsterdam).

De 1995 à 1999 : évolution du langage aux états-unis (CNRI, ...) , **version 1.6 en 1999**

A partir de **2001** et la version 1.6.1 , l'évolution du langage python est contrôlée par la "**Python Software Foundation**".

Python a longtemps existé en version 2.x (durant la décennie 2000-2010) .

Les version 3.x (à partir de 2008/2010) sont sur certains petits points en rupture avec les version 2.x. Il est donc conseillé aujourd'hui de ne plus utiliser l'ancienne version 2 (dans la mesure du possible)

La version **3.7** de python (datant de **2018**) est une bonne version récente et stable du langage python.

1.3. Distributions de "python"

En tant que langage interprété , python peut être pris en charge par plusieurs variantes du moteur d'interprétation.

L'implémentation de référence est **CPython** (interpréteur codé en langage C) .

Il existe aussi **Jython** (basé sur une machine virtuelle java) , ...

En outre, on trouve aujourd'hui certaines distributions packagées de python incluant "moteur d'interprétation + éditeurs + ensemble de bibliothèques + ...) .

Les principales distributions (basées sur CPython) sont :

- **WinPython**
- **Annaconda**
- ...

La plupart des distributions sont à usage scientifique et elles incluent les bibliothèques de calculs **numPy** et **sciPy** .

Autrement dit, pour installer python sur un ordinateur on peut installer que python ou bien toute une distribution telle qu'anaconda .

II - Prise en main de l'interpréteur Python

1. Installation et première utilisation de python

Avant de pouvoir utiliser le langage python il faut installer un des interpréteurs disponibles

1.1. Quelques installations possibles de python

Python est souvent installé d'office sur les distributions linux mais pas forcément dans une version récente . Par exemple sur **linux ubuntu 18.04** , le python 3.6 pré-installé se lance avec la commande "**python3**".

Pour installer python sur un ordinateur windows , on peut se connecter sur le site officiel <https://www.python.org/> et effectuer un téléchargement de l'installateur via l'url <https://www.python.org/ftp/python/3.8.2/python-3.8.2.exe> (environ 26 Mo) .



On peut éventuellement ajouter Python au PATH pour pouvoir ultérieurement le lancer facilement depuis une fenêtre CMD ou autre .

On peut également installer indirectement Python en installant toute une distribution telle qu'anaconda .

<https://www.anaconda.com> ,
<https://repo.anaconda.com/archive/Anaconda3-2020.02-Windows-x86.exe> (*environ 478 Mo*)

Répertoire d'installation par défaut: *C:\ProgramData\Anaconda3*, éventuellement ajouté au PATH.

1.2. Prise en main de l'interpréteur python

Au sein d'une fenêtre de commande (CMD ou PowerShell ou shell linux ou ...) , la commande **python -V** permet d'afficher la version de l'interpréteur python (ex : 3.7.6) .

En lançant la commande "**python**" sans argument , on peut ainsi **lancer l'interpréteur python en mode interactif** . Celui ci nous invite alors à saisir des commandes après une invite (prompt) "**>>>**" .

Toute "commande / ordre / expression" saisi(e) est alors immédiatement interprété(e) et le résultat s'affiche immédiatement .

```
python
Python 3.7.6 .....
>>> 2+3
5
>>> 4*6
24
>>>
```

1.3. Ecriture et lancement d'un programme python

Un fichier de code python a par convention l'extension ".py" et peut être saisi avec un très grand nombre d'éditeur (notepad++, visual studio code , ...) .

Exemple: hello.py

```
a=2
b=3
c=a+b
print('Hello world')
print('c=',c)
```

Pour lancer ce script (petit programme) , on peut lancer la commande

python hello.py

Ce qui provoque l'affichage suivant :

Hello world

c= 5

Et pour automatiser un lancement depuis un double click via l'explorateur de fichiers de windows on peut éventuellement écrire un fichier de lancement tel que celui ci :

lancer_prog_python.bat

```
REM avec PATH contenant le répertoire d'installation de python
python hello.py
pause
```

1.4. Instructions élémentaires du langage python .

NB : En langage python les **commentaires** sont des **fin de lignes commençant par #**

p1.py

```
a=1 # nomVariable=valeur_a_affecter
print(a) # affiche la valeur de la variable a (ici 1)
a=2
print(a) # affiche la nouvelle valeur de la variable a (ici 2)
b=a*3+4 # variableResultat = expression d'un calcul
c=a+b
print("b=",b,"c=",c) # affiche plusieurs choses en les séparant par des espaces
                        # affiche ici b= 10 c= 12
prenom = "alex" # une chaîne de caractères est délimitée en python par des " " ou des ' '
nom = 'Therieur'
nomComplet = prenom + ' ' + nom # concaténation (ajout bout à bout)
print(nomComplet) # affiche alex Therieur

# input('texte question') demande à saisir/renseigner une valeur
age = input ("quel est ton age ? ")
print ('age renseigné: ', age); # affichera la valeur choisie/précisée .
```

Attention : un + entre 2 chaînes de caractères (string) déclenche une concaténation (valeurs juxtaposées bout à bout) tandis qu'un + entre 2 nombres déclenche une addition

La fonction **float()** converti une chaîne de caractères en une valeur numérique (avec potentiellement une virgule notée "." en anglais) .

Exemple :

```
a=input('a:') # exemple a: 2 et a vu comme '2'
b=input('b:') # exemple b: 3 et b vu comme '3'
c=a+b # '2' + '3' = '23'
print('c=a+b=',c) # affiche par exemple c=a+b= 23

a=float(input('a:')) # exemple a: 2 et a vu comme 2.0
b=float(input('b:')) # exemple b: 3 et b vu comme 3.0
c=a+b # 2.0 + 3.0 = 5.0 = 5
print('c=a+b=',c) # affiche par exemple c=a+b= 5
```

Quelques exercices :

Ecrire un petit script calculant la moyenne de 2 nombres avec des valeurs à saisir :

x=4 ou 2 ou ...

y=8 ou 10 ou ...

moyenne= 6.0 ou ...

moyenne.py

Autres essais libres. c'est en forgeant que l'on devient forgeron.

III - Syntaxes élémentaires , types , boucles

1. Titre Section1 du chapitre englobant

Texte

1.1. Titre Paragraphe

Texte

IV - fonctions python

1. Titre Section1 du chapitre englobant

Texte

1.1. Titre Paragraphe

Texte

ANNEXES

V - Annexe – Bibliographie, Liens WEB + TP

1. Bibliographie et liens vers sites "internet"

2. TP