

Python - résumé

1) Syntaxe

#	Pour ajouter des commentaires
indentation	une tabulation ou deux espaces pour matérialiser des blocs d'instructions (structures conditionnelles, boucles et fonctions)

2) Les sorties

Pour afficher du texte ou des nombres, on utilise la fonction **print**.

```
print("Bonjour !")
```

Affichage : Bonjour !

```
Print(2)
```

Affichage : 2

```
a = -3
print(1)
print(2)
print("Le carré de ", a, " est ", a * a)
```

Affichage : 1

2

Le carré de -3 est 9

```
print(1, end = " ") # end = " " évite le retour à la ligne
print(2, end = " ")
print(3, end = " ")
```

Affichage : 1 2 3

3) Les entrées

Lorsqu'un programme demande de saisir des informations au clavier on utilise la fonction **input**.

Saisie d'une chaîne :

```
nom = input("Quel est votre nom ?")
```

Saisie d'un entier

```
n = int(input("Nombre de frères et sœurs"))
```

Saisie d'un nombre à virgule

```
x = float(input("Température à 12h00"))
```

4) Les calculs avec python

+ addition
- soustraction
* multiplication
/ division
** puissance
// division entière
% reste de la division
int partie entière
round valeur approchée

```
a = 2
print(2 + 3) # affichage 5
print(2 - 3) # affichage -1
print(3 * a) # affichage 6
print(15 / 6) # affichage
print(15 // 6) # affichage 2
print(15 % 6) # affichage 3
print(round(3.14159265, 2)) # résultat 3.14
print(round(3.14159265, 4)) # résultat 3.1416
print(int(3.14159265)) # résultat 3
print(int(3.8)) # résultat 3
```

5) Écriture des nombres

```
x = 2 # x est un entier
print(x) # affichage 2

x = 2.0 # x est du type float (nombre à virgule)
print(x) # affichage 2.0
```

```
x = int(7 / 2) # x est un entier
print(x) # affichage 3
```

```
x = float(6 / 2) # x est du type float
print(x) # affichage 3.0
```

```
x = 2.75e3 # écriture scientifique
print(x) # affichage : 2750.0
```

6) Instructions conditionnelles if, else

6.1) Avec un if

```
nombre = float(input("Choisissez un nombre"))
if (nombre > 0):
    print("Le nombre choisi est positif")
print("Fin du programme")
```

6.2) Avec un if et un else

```
nombre = float(input("Choisissez un nombre"))
if (nombre > 0):
    print("Le nombre choisi est positif")
else :
    print("Le nombre choisi est négatif ou nul")
print("Fin du programme")
```

6.3) Les opérateurs de test

Test en python 3	Test si
if (n==0):	n est égal à zéro
if (n>0):	n est positif
if (n!=34):	n est différent de 34
if (n>0) and (n<10):	n est compris strictement entre 0 et 10
If 0< n < 10 :	n est compris strictement entre 0 et 10
if (n<0) or (n>10):	n est négatif ou strictement supérieur à 10
if (n%5==0):	n est divisible par 5

6.4) Différence entre = et ==

```
n = 10 # affectation
print(n) # affichage : 10
print(n != 0) # affichage : True
print(n == 0) # affichage : False
n = 0
print(n== 0) # affichage : True
```

6.5) Tester plusieurs valeurs d'une variable

(remplace le case d'autres langages)

```
n = 10
if n < 0 :
    print("Le nombre est négatif")
elif n == 0 :
    print("Le nombre est égal à zéro")
else :
    print("Le nombre est positif")
```

Affichage : Le nombre est positif

7) La boucle for ... in : boucle pour

```
for n in range(3):
    print("Bonjour")
print("Fin")
```

Résultat : Bonjour

Bonjour
Bonjour
Fin

Remarque :

- toutes instructions indentées font partie de la boucle
- range(3) définit 3 itérations

```
for n in range(10):
    print(n)          # affichage des valeurs de 0 à 9
```

Remarques :

- toutes instructions indentées font partie de la boucle
- range(10) définit 10 itérations mais la première valeur est 0 donc la dernière est 9

```
for n in range(1,10):
    print(n)          #affichage des valeurs de 1 à 9
```

```
n = 10
for i in range(1,n + 1):
    print(n)          #affichage des valeurs de 1 à n
```

8) Boucle while : boucle tant que

8.1) Nombre d'itérations connu à l'avance

Exemple de deux programmes équivalents

```
for n in range(10):
    print(n)
print("Fin du programme")
```

```
n=0
while n<10:
    print(n)
    n=n+1
print("Fin du programme")
```

8.2) Nombre d'itérations non connu à l'avance

```
gagne = 0
while gagne == 0:
    n=int(input("Entrer un nombre"))
    if n == 10 :
        gagne = 1
print("Gagné")
```

9) Les chaînes de caractères

9.1) Initialisation d'une variable

```
txt="Bonjour !"
print(txt)
```

Résultat : Bonjour !

```
txt='Bonjour !'
print(txt)
```

Résultat : Bonjour !

Pour aller plus

```
txt="Bonjour !", dit-elle.\n"Bonjour", répondit-il.'
print(txt)
```

Résultat : "Bonjour !", dit-elle.

"Bonjour", répondit-il.

Remarques :

- \n insère un retour à la ligne.
- \' insère une apostrophe dans une chaîne délimitée par des apostrophes.
- De même, \" insère des guillemets dans une chaîne délimitée par des guillemets.

9.2) Accès aux caractères individuels d'une chaîne

```
ch = "Constance"
print(ch[0], ch[4], ch[8])
```

Résultat : Cte

9.3) Opérations sur les chaînes

```
a = "Un cours"
b = " ça s'apprend !"
c = a + b
print(c)          # affichage : Un cours ça s'apprend !
```

Remarque : le + réalise la concaténation de deux chaînes

```
a = "123"
b = "321"
c = a + b
print(c)          # affichage : 123321
```

```
ch = "Pierre"
print(len(ch))    # affichage : 6, longueur de la
                  # chaîne
```

```
ch = "12"
n = int(ch)
print(n + 8)      # affichage : 20
```

```
a = 12             # a est du type nombre
b = str(a)         # b est du type chaîne
print(b)           # affichage : 12
```

10) Les listes

Une liste est une collection d'éléments séparés par des virgules, l'ensemble étant enfermé dans des crochets.

```
liste1 = ["a", "b", 20, 30, "cd"]
print(liste1[0], liste1[2], liste1[4]) #affiche a20cd

liste1[0] = "e"
print(liste1) #affiche ["e", "b", 20, 30, "cd"]

print(len(liste1)) #affiche 5

del(liste1[3])
print(liste1) #affiche ["e", "b", 20, "cd"]

liste1.append("a")
print(liste1) #affiche ["e", "b", 20, "cd", "a"]

print(liste1.index("b")) #affiche 1

liste1.remove("b")
print(liste1) #affiche ["e", 20, "cd", "a"]
```

Des listes avec range

range(5) est une liste pré-remplie de 5 éléments : [0, 1, 2, 3, 4]
autres syntaxes :
range(2,5) ↔ [2, 3, 4]
range(0,5,2) ↔ [0, 2, 4]
(la syntaxe est range(start, stop, step))

11) Les fonctions et les procédures

11.1) Les fonctions

Un fonction admet un ou plusieurs arguments et retourne une valeur.

```
def cube(x): #Définition de la fonction cube
    y = x**3
    return y

a = cube(2) # Appel de la fonction cube
print(a) #affichage 8
```

```
def puissance(x,n): #définition de la fonction
    y = x**n
    return y
```

```
print(puissance(3,2)) # ppeel de la fonction
#affichage 9
```

11.2) Les procédures

Une fonction qui n'admet pas d'argument ou qui ne retourne pas de valeur est appelée procédure.

```
def table7(): #Définition de la fonction
    for i in range(1,10+1):
        print(7 * i) #La fonction ne retourne rien

table7() #Appel sans argument
#Résultat : 7 14 21 28 35 42 49 56 63 70
```

```
def table(base): #Définition de la fonction
    for i in range(1,10+1):
        print(base * i) #La fonction ne retourne rien

table(8) #Appel avec un argument
#Résultat : 8 16 24 32 40 48 56 64 72 80
```

11.3) Variable globale, variable locale

Le même nom mais deux variables différentes

```
a = 15 #variable globale
def plusUn():
    a = 0 #variable locale (à la fonction)
    a = a + 1
    print(a)

plusUn() #affiche 1
print(a) #affiche 15
```

Programme qui ne fonctionne pas

```
a = 15 #variable globale
def plusUn():
    a = a + 1 #Erreur car a n'est pas initialisée
```

```
print(a)

plusUn()
print(a)
```

Utilisation d'une variable globale dans une procédure

```
a = 15 #variable globale
```

```
def plusUn():
    global a #on utilise la variable globale a
    a = a + 1
    print(a) #affichage 16

plusUn() #appel de la fonction, affichage 16
print(a) #affichage : 16
#la nouvelle valeur de a
```

12) Déclaration des bibliothèques

Méthode 1 : rappel du nom de la bibliothèque obligatoire
import math
x = math.sqrt(25)

```
import random
x = random.random()
```

Méthode 2 : pas de rappel du nom de la bibliothèque

```
from math import *
x = sqrt(25)
```

```
from random import *
x = random()
```

13) Fonctions mathématiques

Fonctions mathématiques (sqrt(x), cos(x), sin(x), pi ...)

```
from math import *
print(sqrt(16))      #affichage : 4.0
print(pi)            #affichage : 3.141592653589793
print(sin(pi/2))     #affichage : 1.0
print(cos(pi/4))n    #affichage :
                    # 0.7071067811865476
```

14) Fonctions aléatoires

```
from random import *
print(random())      #nombre aléatoire dans [0;1[
                    #exemple :
                    # 0.8365827741773375
```

```
L = [7, 3, 8, 5, 6]
print(choice(L))     #Choisir un élément
                    #dans une liste1
                    #Exemple : 5
```

```
liste1 = ['Pierre', 'Paul', 'Jacques']
print(choice(liste1)) #Choisir un élément
                    # dans liste1
                    #Exemple : Jacques

print(randint(5, 10)) #Entier aléatoire avec
                    # 5 <= N <= 10
                    #Exemple : 7
```

15) Graphiques avec la bibliothèque matplotlib

Au minimum

```
import matplotlib.pyplot as plt

plt.plot(x,y,"ro") #afficher un point rouge et rond
plt.show()        #terminer et montrer le
                    # graphique
```

Avec plus d'options

```
import matplotlib.pyplot as plt

plt.plot(x,y,"ro") #afficher un point rouge et rond
plt.title("Titre du graphique")
plt.axis([xmin, xmax, ymin, ymax]) # Fenêtre
                                    # d'affichage
plt.grid(True)                    # afficher la grille
plt.ylabel(" Texte pour les ordonnées")
plt.xlabel("Texte pour les abscisses")
plt.show()
```

Formes et couleurs des points

(nombreuses possibilités voir site officiel)

r rouge	o rond
g vert	^ triangle
b bleu	s carré

16) Ressources

16.1) Sites du lycée

(documentation, logiciels à télécharger)

<http://www.isnaborda.free.fr/>
<http://www.borda.free.fr/>

16.2) Documentation sur matplotlib

Matplotlib tutorial - LaBRI

(nombreux exemples, liste des options)

<https://www.labri.fr/perso/nrougier/teaching/matplotlib/>

Site officiel : matplotlib.org

<https://matplotlib.org/>

16.3) Documentation sur python

Wikibook : Programmation Python

https://fr.wikibooks.org/wiki/Programmation_Python

Python au lycée 1

<http://python.lycee.free.fr/>

Python au lycée 2

<http://site.ac-martinique.fr/mathematiques/wp-content/uploads/2014/06/IntroPython.pdf>

Apprendre-python.com

<http://apprendre-python.com/>

Apprendre Python en ligne

<https://www.codecademy.com/fr/learn/python>

16.4) Programmation en ligne

Avec la console

<https://repl.it/languages/python3>

Avec la console et matplotlib

<https://trinket.io/features/python3>