



Commandes de base Python



L'instruction `print` affiche une information à l'écran :

instruction	Equivalent en français	Résultat à l'écran
<code>print("Bonjour")</code>	Affiche à l'écran la chaîne de caractère entre les guillemets	Bonjour
<code>Bonjour = 5</code> <code>print(Bonjour)</code>	Affecte la valeur 5 à la variable <code>Bonjour</code> Affiche cette variable	5
<code>a = 3</code> <code>print("Il est",a,"h")</code>	Affecte la valeur 3 à la variable <code>a</code> Affiche du texte et la variable <code>a</code>	Il est 3 h
<code>a = 3.526205</code> <code>print("a =",round(a,2))</code>	Affecte la valeur 3,526205 à la variable <code>a</code> Affiche du texte et la variable <code>a</code> arrondie à deux chiffres après la virgule	a = 3.53

L'instruction `input` attribue à une variable un contenu :

<code>a = input("Quel est ton nom ?")</code>	L'instruction attribue à la variable <code>a</code> la chaîne de caractères que rentrera l'utilisateur
<code>age = int(input("Quel est ton age?"))</code>	La commande <code>int</code> impose à l'utilisateur d'entrer un entier naturel.
<code>rayon = float(input("R = ? "))</code>	La commande <code>float</code> impose à l'utilisateur d'entrer un nombre qui sera considéré comme un nombre décimal.
<code>nom = str(input("Nom = ? "))</code>	La commande <code>str</code> fait que le contenu entré par l'utilisateur sera considéré comme une chaîne de caractères.

L'instruction `if` effectue un test logique :

<code>if i == 3 :</code> <code> print i</code>	Affiche la valeur de <code>i</code> à la condition que <code>i</code> soit égal à 3
<code>if i != 3 :</code> <code> print ("mauvaise réponse")</code>	Affiche mauvaise réponse si <code>i</code> est différent de 3
<code>if age > 17 :</code> <code> print ("Majeur")</code> <code>else :</code> <code> print ("mineur")</code>	Affiche Majeur si la variable <code>age</code> est supérieure à 17, sinon (else) affiche Mineur

L'instruction `for` effectue une boucle. Elle répète tout le contenu indenté placé sous elle :

instruction	Equivalent en français	Résultat à l'écran
<code>for i in range(5,9) :</code> <code> print i</code>	Pour <code>i</code> allant de 5 à 8 Afficher <code>i</code>	5 6 7 8
<code>for i in range(3) :</code> <code> print i</code>	Pour <code>i</code> allant de 0 à 2 Afficher <code>i</code>	0 1 2
<code>for i in range(2) :</code> <code> print ("réponse",i)</code> <code> print ("*****")</code>	Pour <code>i</code> allant de 0 à 1 Afficher réponse <code>i</code> Afficher *****	réponse 0 ***** réponse 1 *****
<code>for i in range(2) :</code> <code> print ("réponse",i)</code> <code>print ("*****")</code>	Pour <code>i</code> allant de 0 à 1 Afficher réponse <code>i</code> Afficher *****	réponse 0 réponse 1 *****

L'instruction **while** effectue une ou un ensemble de lignes indentées tant que :

instruction	Equivalent en français	Résultat à l'écran
<code>i = 0</code>	Donne à i la valeur 0	1
<code>while i < 4 :</code>	Tant que i est plus petit que 3 faire	2
<code>i = i + 1</code>	Ajouter 1 à i	3
<code>print (i)</code>	Afficher i	4

L'instruction **import** appelle une bibliothèque contenant de nouvelles instructions indisponibles au départ :

<code>import math</code> <code>print (math.pi)</code> <code>print (math.sin(3))</code>	Affiche à l'écran la valeur de π qui se trouve dans la bibliothèque math ainsi que la valeur de $\sin(3)$.
<code>import math as m</code> <code>print (m.pi)</code>	Renomme la bibliothèque math en m .
<code>import random as rnd</code> <code>print (rnd.randint(2,5))</code>	Affiche à l'écran un entier naturel pris au hasard entre 2 et 5
<code>import time as ti</code> <code>date = int(ti.time())</code> <code>print (date)</code>	Importe la bibliothèque time et place dans la variable date la date actuelle en secondes.

Les opérateurs élémentaires :

instruction	Equivalent en français	Résultat à l'écran
<code>a = 3</code> <code>b = 2</code> <code>print ("somme : ",a+b)</code> <code>print ("produit : ",a*b)</code> <code>print ("division : ",a/b)</code> <code>print ("puissance : ",a**b)</code>	Affecte à a la valeur 3 Affecte à b la valeur 2 Affiche la somme de 2+3 Affiche le produit de 2×3 Affiche la division de 2 par 3 Affiche 3 à la puissance 2	somme : 5 produit : 6 division : 1.5 puissance : 9
<code>a = "3"</code> <code>b = "2"</code> <code>print (a+b)</code>	Affecte à a le caractère 3 Affecte à b le caractère 2 Affiche la concaténation de 2 et 3	32

Tracé de graphique avec la bibliothèque (ou module) **matplotlib.pyplot** :

<code>import matplotlib.pyplot as plt</code> <code>plt.plot(2,5)</code> <code>plt.show()</code>	Importe la bibliothèque matplotlib.pyplot et la renomme en plt . Place dans un graphe le point d'abscisse x = 2 et d'ordonnée y = 5 Affiche le graphe.
<code>plt.plot([2,4,6],[4,3,7])</code> <code>plt.show()</code>	Place sur un graphe les points de coordonnées (2 ;4), (4 ;3) et (6 ;7) Affiche le graphe.
<code>plt.plot([2,4,6],[4,3,7], "g+")</code> <code>plt.xlim(0,10)</code> <code>plt.ylim(-5,10)</code> <code>plt.show()</code>	Affiche des points en forme de croix vertes (g = green). L'axe des x commence à 0 et finit à 10 L'axe des y commence à - 5 et finit à 10
<code>plt.plot([2,4,6],[4,3,7], "cs")</code> <code>plt.xlabel("temps")</code> <code>plt.ylabel("position")</code> <code>plt.title("évolution de la voiture")</code> <code>plt.show()</code>	Affiche des points en forme de carré (s = square) cyan (c = cyan). Affiche le mot temps en dessous de l'axe des x Affiche le mot position en dessous de l'axe des x Affiche le titre évolution de la voiture en dessus du graphe.

Couleurs disponibles :

b : blue
g : green
r : red
c : cyan
m : magenta
y : yellow
k : black
w : white

Forme des points

. point
, pixel
o rond
< ^ > v triangle
s carré
p pentagone
* étoile
+ croix
x croix

il en existe encore d'autres...