

## 00 La programmation en langage Python

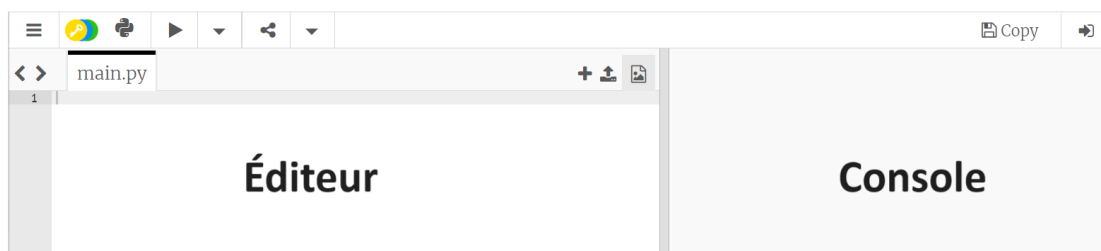
### 00-01 L'éditeur et la console

#### Définitions

La rédaction d'un programme informatique s'effectue dans l'**éditeur** et le résultat apparaît dans la **console**.

#### Application

Aller sur le site <https://trinket.io/python3> .



Écrire les commandes suivantes dans l'éditeur puis observer le résultat dans la console.

Éditeur	Console
<pre>print("Hello world !")</pre>	
<pre>print(3+2*4) print(10/2-1)</pre>	
<pre>print(3**2) print(43//4) print(43%4)</pre>	
<pre>print(min(2,3,5,7)) print(round(2.718,1))</pre>	

#### Définition

Le symbole **%** se nomme **modulo** et il n'a rien à voir avec les pourcentages : il permet d'obtenir le reste d'une division euclidienne.

## 00-02 Les variables

### Définition

Une **variable** est un mot auquel on affecte une valeur.

### Applications

Éditeur	Console
<pre>a,b,c,d = 1,2,"tic","tac" print(a+b) print(c+d)</pre>	

### Définitions

Les variables peuvent avoir différents **types**. Les plus répandus sont :

- les nombres entiers (**int**)
- les nombres à virgule (**float**)
- les chaînes de caractères (**str**)

### Applications

Éditeur	Console
<pre>a=2 print(type(a)) print(type("a")) print(type(a/1))</pre>	

### Propriété

On ne peut pas utiliser les mots réservés suivants pour nommer des variables :

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

## 00-03 Modification du contenu d'une variable

Éditeur	Console
a=3 a=a+2 print(a) a+=2 print(a)	
a=3 a*=2 print(a) a**=2 print(a)	

### Définitions

Une **variable booléenne** ne peut prendre que deux valeurs : **True** ou **False**.

Deux symboles d'égalité == forment un **test d'égalité** qui est soit **True** soit **False**.

### Applications

Éditeur	Console
print(4>3) print(4<3) print(4<=3) print(4==3) print(4!=4)	
print(not True) print(True==0) print(True==1) print(True==2) print(False==0) print(False==1) print(False==2)	
print((2==2) and (3==3)) print((2==3) and (3==3)) print((2==2) or (3==3)) print((2==3) or (3==3))	

## 00-04 Les boucles

### Définitions

Les **boucles** permettent de répéter une série d'instructions.

On utilise la boucle **for** lorsque l'on connaît le nombre exact d'itérations de la boucle.

On utilise la boucle **while** lorsque l'on connaît la condition de fin d'exécution.

### Applications

#### 1. La boucle **for**

Éditeur	Console
<pre>for x in range(5):     print(x)</pre>	
<pre>for x in range(5):     print(x)     print("Fin !")</pre>	
<pre>for x in range(5):     print(x)     print("Fin !")</pre>	
<pre>for x in range(2,5):     print(x)</pre>	

#### 2. La boucle **while**

Éditeur	Console
<pre>i = 0 while i &lt; 5:     print(i)     i=i+1</pre>	
<pre>i = 0 while True:     print(i)     i=i+1</pre>	

## 00-05 Les conditions

### Définitions

Les **conditions** permettent d'adapter un programme à des cas différents.

Elles commencent par **if**.

Elles peuvent contenir un sous-cas commençant par **else**.

Elles peuvent contenir plusieurs sous-cas commençant par **elif** (contraction de else if).

### Applications

Éditeur	Console
<pre>if True and True :     print("Pim") if True and False :     print("Pam") if True or False :     print("Pom")</pre>	
<pre>if 10**2==100 :     print("Gagné") else :     print("Perdu")</pre>	
<pre>a= 11**3 if a==1000 :     print("Gagné") elif a&lt;1000 :     print("Trop petit") else :     print("Trop grand")</pre>	
<pre>a=9**3 if a==1000 :     print("Gagné") elif a&gt;0 :     print("positif") elif a&lt;1000 :     print("Trop petit") else :     print("Trop grand")</pre>	

## 00-06 Les fonctions

### Définitions

Une **fonction** permet d'effectuer une tâche précise dans un programme long.  
Une fonction commence par **def** et se termine par **return**.

### Applications

Éditeur	Console
<pre>def boule() :     volume=round(4/3*3.14*rayon**3,0)     return volume  rayon=1 print(boule())</pre>	
<pre>def boule(rayon) :     volume=round(4/3*3.14*rayon**3,2)     return volume  print(boule(2))</pre>	
<pre>def boule(rayon) :     volume=round(4/3*3.14*rayon**3,2)     return volume  rayon=10 print(boule(2))</pre>	
<pre>def essai(a) :    # a est un paramètre     a=0           # a est variable locale     a=a+1     return a  a=10              # a est variable globale print(essai(5)) print(a)</pre>	

## 00-07 Communiquer avec l'utilisateur

### Définition

L'instruction **input** permet d'affecter à une variable une chaîne de caractère entrée par l'utilisateur.

### Applications

Éditeur	Console
<pre>a = input("Votre âge ?") print("Vous avez", a,"ans") print(type(a))</pre>	
<pre>a = input("Votre âge ?") print("Vous aurez", a+1,"ans")</pre>	
<pre>a = int(input("Votre âge ?")) print("Vous aurez", a+1,"ans") print(type(a))</pre>	

Éditeur	Console
<pre>print("J'aime le pain") print('Tu aimes le pain')</pre>	
<pre>print('J'aime le pain')</pre>	
<pre>print('J\'aime le pain')</pre>	
<pre>print("Paul a ", 12, " ans") print("\nPaul a ", 12, " ans") print("Paul a\t ", 12, " ans") print("Paul a ", 12, " ans", sep="-")</pre>	

## 00-08 Les listes

### Définition et notation

Une **liste** (ou **tableau**) est une variable contenant plusieurs éléments, éventuellement de différentes natures, placés **entre crochets** et **séparés par des virgules**.

### Applications

Éditeur	Console
<pre>a=[1,2,3] a=a+[5] print(a) print(2*a)</pre>	
<pre>golf=["parcours",18, True] print(golf[1]) h=["Hugo",1802,1885] auteurs=[golf,h] print(auteurs[1][2])</pre>	
<pre>a=[3,7,8,2,1,4] print(len(a)) print(a.count(2)) print(a.index(2))</pre>	
<pre>a=[3,7,8,2,1,4] a.pop(1) print(a) a.append(17) print(a) a.insert(1,20) print(a)</pre>	
<pre>a=[3,7,8,2,1,4] copie=a a.sort() print(a) print(copie)</pre>	
<pre>a=[3,7,8,2,1,4] copie=a a=sorted(a) print(a) print(copie)</pre>	



## 00-09 Les modules

### Définitions

Les **modules** sont des ensembles de variables et de fonctions prédéfinies.  
Pour utiliser un module dans un programme, il faut d'abord l'**importer**.

### Applications

Éditeur	Console
<code>print(pi, "vaut environ", 3.14)</code>	
<code>from math import* print(pi, "vaut environ", 3.14)</code>	
<code>from math import* print(sqrt(4)) print(floor(2.718)) print(ceil(2.718))</code>	

Éditeur	Console
<code>from random import* print(random())</code>	
<code>from random import* print(randint(5,10))</code>	
<code>from random import* print(sample([2,3,5,7], 2))</code>	

### Remarques

La liste des fonctions contenues dans les deux modules ci-dessus se trouve ici :

- math : <https://www.programiz.com/python-programming/modules/math>
- random : <https://www.programiz.com/python-programming/modules/random>

Un troisième module est utilisé pour tracer des graphiques.

Il se nomme **matplotlib.pyplot** et sera l'objet du prochain (et dernier) chapitre de ce cours.

## 00-10 Le module graphique

### Propriété

On trace des graphiques à l'aide du module **matplotlib.pyplot**.

### Applications

Éditeur	Console
<pre>from matplotlib.pyplot import * grid() show()</pre>	
<pre>from matplotlib.pyplot import * axis([-10,10,0,20]) grid() show()</pre>	
<pre>from matplotlib.pyplot import * axis([-10,10,0,20]) title("Mon joli graphique") xlabel("Abscisses") ylabel("Ordonnées") show()</pre>	

Éditeur	Console
<pre>from matplotlib.pyplot import * scatter(1,2) scatter(3,4,marker='*') scatter(5,6,marker='X') show()</pre>	
<pre>from matplotlib.pyplot import * x=[1,2,3,4,5] y=[1,4,9,16,25] plot(x,y) show()</pre>	