

00 La programmation en langage Python

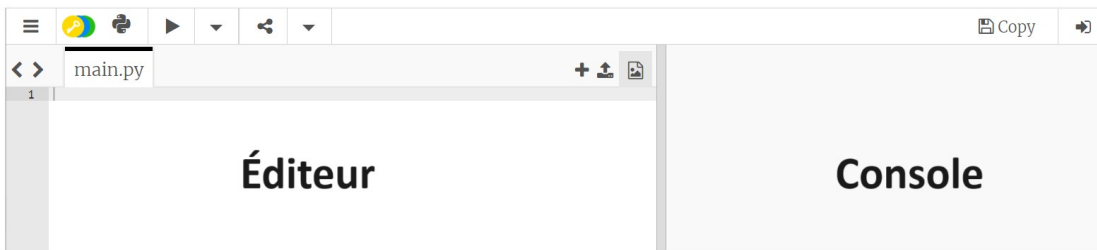
00-01 L'éditeur et la console

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 % n'a rien à voir avec les pourcentages : il se nomme **modulo** et 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 Contenu d'une variable

Éditeur	Console
<pre>a=3 a=a+2 print(a) a+=2 print(a)</pre>	
<pre>a=3 a*=2 print(a)</pre>	

Définitions

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

La combinaison de symboles **==** est un **test d'égalité** qui est soit **True** soit **False**.

La combinaison de symboles **!=** est un test de **non égalité** qui est soit **True** soit **False**.

Applications

Éditeur	Console
<pre>print(4>3) print(4<3) print(4<=4)</pre>	
<pre>print(4==3) print(4!=3) print(4!=4)</pre>	
<pre>print((2==2) and (3==3)) print((2==3) and (3==3)) print((2==2) or (3==3)) print((2==3) or (3==3))</pre>	

Exercice

Afficher « True » ou « False » selon qu'un entier donné est multiple ou pas d'un autre entier.

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

2. La boucle **while**

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

Exercices

1. Afficher les n premiers multiples de a . Tester avec $n = 10$ et $a = 59$.
2. Afficher les multiples de a inférieurs à b . Tester avec $a = 59$ et $b = 1000$.
3. Pour a et b donnés, déterminer le plus grand multiple de a inférieur ou égal à b .
Tester avec $a = 59$ et $b = 1000$.

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** (« else if »).

Applications

Éditeur	Console
<pre>if True : print("Tic") if False : print("Tac") 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>note = 18 if note == 20 : print("Parfait") elif note > 15 : print("Très bien") elif note >= 10 : print("Correct") else : print("Insuffisant")</pre>	

Exercice

Afficher tous les diviseurs d'un nombre n donné. Tester avec $n = 84$.

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** suivi du résultat renvoyé.

Applications

Éditeur	Console
<pre>def boule(): volume=round(4/3*3.141593*r**3,0) return volume r=2 print(boule())</pre>	
<pre>def boule(r): volume=round(4/3*3.141593 *r**3,2) return volume print(boule(2))</pre>	
<pre>def test(a): # a est un paramètre b=0 # b est une variable locale a=a+1 return a a=10 # a est une variable globale b=1 # b est une variable globale test(4) print(test(5)) print(a,b)</pre>	

Exercice

1. Écrire la fonction **premier(*n*)** qui renvoie « True » si le nombre entier *n* est premier et « False » sinon.
2. Utiliser cette fonction pour savoir combien il y a de nombres premiers entre 100 et 200.

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("\nPaul a ", 12, " ans")</pre>	
<pre>print("Paul a\t ", 12, " ans")</pre>	
<pre>print("Paul a ", 12, " ans", sep="-")</pre>	

Exercice

Écrire la fonction **encadre_racine(a,n)** qui affiche l'encadrement décimal de la racine carrée de a avec la précision de n chiffres après la virgule.

Exemple avec $a = 2$ et $n = 5$: « La racine carrée de 2 est comprise entre 1,41421 et 1,41422 ».

00-08 Les listes

Définition et notation

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

Applications

Éditeur	Console
<pre>c=["Corneille",1606,1684] print(c[1]) h=["Hugo",1802,1885] auteurs=[c,h] print(auteurs[1][2])</pre>	
<pre>a=[5,6,7] a=a+[8] print(a) a.append(9) print(a)</pre>	
<pre>a=[3,7,8,2,1,4] print(len(a)) print(a.count(2)) print(a.index(2))</pre>	
<pre>a=[3,2,1,4] copie=a a.sort() print(a,copie)</pre>	
<pre>a=[3,2,1,4] copie=a a=sorted(a) print(a,copie)</pre>	

Exercice

Deux nombres premiers sont « jumeaux » lorsque leur différence vaut 2. Exemple : 3 et 5.
Écrire la fonction **jumeaux(*m*,*n*)** qui renvoie le nombre de couples de nombres premiers jumeaux compris entre les entiers *m* et *n*.

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

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

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>

Exercice

Écrire la fonction **des(*n*,*d*)** qui renvoie une liste simulant le lancer de *n* dés à *d* faces.

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 * 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>	

Exercice

Tracer la courbe représentative de la fonction carré.