

Programmation en Python

Les opérateurs mathématiques

Ils existent plusieurs opérateurs pour effectuer des calculs dans vos scripts Python :

Opérations	Symboles	Exemples
Addition	+	2 + 5 donne 7
Soustraction	-	8 – 2 donne 6
Multiplication	*	6 * 7 donne 42
Puissance	**	5 ** 3 donne 125
Division	/	7 / 2 donne 3,5 (en Python 3)
Modulo	%	7 % 2 donne 1 (reste de la division entière)
Quotient	//	7 // 2 donne 3 (quotient de la division entière)

Cependant certains opérateurs n’existent pas dans le langage Python natif, et on doit les ajouter préalablement avant de les utiliser en important des bibliothèques.

```
# Importation de la bibliothèque math
import math

# racine carré de 9
racine = math.sqrt(9)
```

Exercice 5

Il est possible d’utiliser ces opérateurs sur des entiers (int) et des flottants (float), mais peut on les utiliser sur des chaînes de caractères ? 🤔

Essayez cette ligne de code Python en remplaçant l’opérateur entre ces deux chaînes de caractères :

```
print( "J’aime la" OPERATEUR "SNT" )
```

Les booléens

Propriété

Un **booléen** est un type de donnée en informatique et en mathématiques qui ne peut prendre que **deux valeurs possibles** :

Vrai (True) ou **Faux** (False).

Il sert à représenter une vérité logique ou le résultat d’une comparaison. Par exemple :

- 2 > 1 est **vrai** → booléen = True
- 5 = 3 est **faux** → booléen = False

Exercice 6

Sur feuille, puis vérifier sur l'ordinateur

1) Donner la valeur (True ou False) des expressions suivantes :

```
15 <= 20 or 1 > 150
2 < 4 and 2 < 3
"A" == "A" and "B"=="B"
"A" != "C"
not (1 < 3)
not (15 <= 20) or 1 < 150
3 < 5 and ((7 < 5) or (2 < 3))
```

2) Que font les opérateurs <= ? > ? == ? and ? or ? not ?

Opérateur	Description

Propriété

La structure conditionnelle "if"

En Python, voici la structure :

```
if condition :
    instruction(s) à effectuer dans la cas où la condition est remplie
else :
    instruction(s) à effectuer dans la cas contraire
```

ATTENTION ! Le bloc `else` n'est pas obligatoire !

Remarque : Le symbole `:` est très important en Python car il marque le début d'un bloc. C'est l'indentation (=décalage) qui délimite le bloc d'instructions.

Exercice 7

Écrire ce code et exécuter le :

```
a=float(input("Entrer un nombre réel : "))
if a>=0 :
    print("Vous avez entré un nombre positif ou nul",a)
else :
    print("Vous avez entré un nombre négatif",a)
```

1) Qu'affiche le programme de l'exemple dans chacun des cas suivants :

1.Avec `a=8` ?

2.Avec `a=-6` ?

3.Avec `a=0` ?