



Langage Python

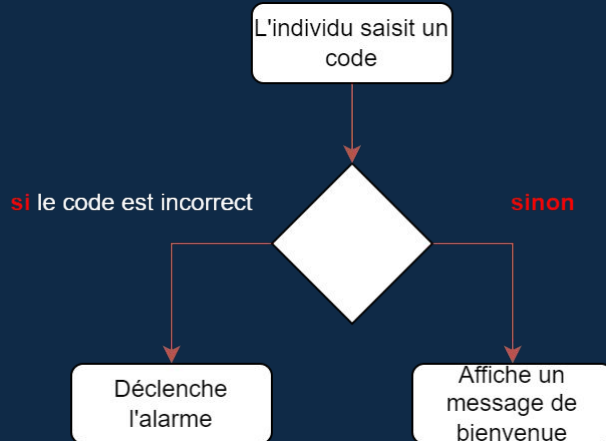
Les bases du langage 2 : Les conditions

Formation POEC Cybersécurité
Théo Hubert

Syntaxe

◆ Comparaisons & Tests

“Lorsque vous concevez un programme, il peut arriver que vous vouliez que votre **programme fasse des choix**. C' est à dire que certaines instructions soient exécutées selon des conditions booléennes. Si telle situation est vraie, alors exécutez cette ou ces instructions. Sinon, exécutez plutôt celle ou celles-là. Pour y arriver, on utilise des **structures conditionnelles**.



Syntaxe

◆ Comparaisons & Tests

Pour utiliser une structure conditionnelle avec Python on utilise :

- une indentation (espace qui indique à Python les instructions qui font partie de la condition)
- un des mots-clés suivants : **if**, **else**, **elif**.

```
nbr = 10

if nbr == 10:
    print("Le nombre est bien 10")
```

Le nombre est bien 10

**Attention à
l'indentation**

```
nbr = 10

if nbr == 10:
    print("Le nombre est bien 10")
    print("Le nombre est bien 10")

File "<ipython-input-29-00049b016624>", line 4
    print("Le nombre est bien 10")
    ^
IndentationError: expected an indented block
```

Syntaxe

◆ Comparaisons & Tests

Fonctionnement d'un test :

La condition d'un test fonctionne à partir de **booléens** :

- Si la condition est vrai => True
- Si la condition est fausse => False

```
[3] age = 15  
    condition = (age <= 18)  
    print(condition)
```

True

```
[4] age2 = 25  
    condition2 = (age2 > age)  
    print(condition2)
```

True

Syntaxe

◆ Tests

Tests à plusieurs cas :

Pour réaliser des tests à plusieurs cas on utilise les conditions **else** et **elif**.

```
nbr = int(input("Veuillez saisir un nombre :"))

if nbr == 0 :
    print("Le nombre saisi est zéro")
elif nbr > 100 :
    print("Le nombre est supérieur à 100")
elif nbr > 50 :
    print("Le nombre est supérieur à 50")
else:
    print("Le nombre est compris entre 1 et 50")
```

Veuillez saisir un nombre :10

Le nombre est compris entre 1 et 50

Syntaxe

◆ Tests

Importance de l'indentation :

Il faut être particulièrement vigilant à **l'indentation** lorsque l'on utilise des tests avec Python.

```
[1] nb = 5
    if nb == 5:
        print("Le test est vrai")
        print(f"car la variable nb vaut {nb}")
```

```
Le test est vrai
car la variable nb vaut 5
```

```
[2] if nb == 5:
    print("Le test est vrai")
    print(f"car la variable nb vaut {nb}")
File "<ipython-input-2-442659a5eea3>", line 3
    print(f"car la variable nb vaut {nb}")
    ^
IndentationError: unindent does not match any outer indentation level
```

SEARCH STACK OVERFLOW

Syntaxe

◆ Tests

Tests multiples :

Il est possible de réaliser des tests multiples en utilisant les opérateurs logiques **and** et **or**

```
a = 2
b = 2
c = 4

if a == 2 and b == 2:
    print("OK")

if c == 2 or a == 2:
    print("OK")

if a != 1 and b == 2 and c >= 2:
    print("a est différent de 1, b est égal à 2 et c est supérieur ou égal à 2")
```

OK
OK
a est différent de 1, b est égal à 2 et c est supérieur ou égal à 2

Syntaxe

◆ Tests

Tests multiples :

Opérateur logique OR

Condition 1	Opérateur	Condition 2	Résultat
Vrai	OU	Vrai	Vrai
Vrai	OU	Faux	Vrai
Faux	OU	Vrai	Vrai
Faux	OU	Faux	Faux

Opérateur logique AND

Condition 1	Opérateur	Condition 2	Résultat
Vrai	ET	Vrai	Vrai
Vrai	ET	Faux	Faux
Faux	ET	Vrai	Faux
Faux	ET	Faux	Faux

Syntaxe

Travaux Pratique

Références

<https://courspython.com/boucles.html>

<https://python.iutsf.org/lecon-4-les-boucles/>

<https://pythontutor.com/visualize.html#mode=display>

<https://courspython.com/fonctions.html>

https://python.developpez.com/cours/apprendre-python3/?page=page_11

<https://www.pierre-giraud.com/python-apprendre-programmer-cours/module-paquet/>

http://sosoftware.free.fr/ExosPython_CorrectionF.php

Fin

*La suite :
Programmation Orienté Objet*

