

INTRODUCTION A LA PROGRAMMATION PYTHON

MatheX – Licence CC BY-NC-SA 4.0 - <https://www.mathexien.com>

#3. Condition

Objectifs:

- Introduire le concept de condition
- Programmer des alternatives

On a souvent besoin dans un programme de n'exécuter une partie du programme que si certaines conditions sont réunies, on utilise pour cela des **conditions**.

En Python, on utilise l'instruction `if` avec la syntaxe suivante:

```
1  if boolean :
2      instruction 1
3      instruction 2
4      ...
5      instruction n
6  instruction_hors_du_if
```

NB:

L'**indentation** fait partie de la syntaxe en Python.

Le bloc d'instructions du `if` est indenté par rapport au `if` (instructions 1 à n)

Quand on revient à l'indentation du `if`, on sort du `if` (instruction_hors_du_if)

```
1  if True:
2      print("OK")
3      print("la condition du if est True")
4      print('on exécute le bloc d\'instructions')
5  print("on est sorti du if True:")
6  if False:
7      print("KO")
8      print("la condition du if est False")
9      print('on n\'exécute pas le bloc d\'instructions')
10 print("on est sorti du if False:")
```

Le booléen dans la condition peut résulter d'une opération qui renvoie un booléen.

Nous avons vu en #2 les opérations logiques (`and`, `or`, `not`)

Il y a aussi les opérations de comparaison qui renvoie un booléen:

- supérieur, inférieur, supérieur ou égal, inférieur ou égale: `>`, `<`, `>=`, `<=`
- égal: `==` (attention un simple égal est une affectation pas une comparaison)
- différent: `!=`

```

1  a = int( input('entrez le nombre entier a:'))
2  b = int( input('entrez le nombre entier b:'))
3
4  if a == b:
5      print('a égal à b')
6  if a != b:
7      print('a est différent de b')
8  if not (a == b):
9      print('a est différent de b')
10 if a < b:
11     print('a est inférieur à b')
12 if a > b and a > 10:
13     print('a est supérieur à b et à 10')

```

Le code ci-dessous, bien que correct, n'est pas optimal car des conditions sont testées inutilement.

On préférera utiliser des **alternatives**.

En Python, la structure est :

```

1  if condition1:
2      instruction_1
3  elif condition2: # elif = else if : la condition2 n'est pas testée si la condition1 était True
4      instruction_2
5  elif condition3: # la condition3 n'est pas testée si une des conditions précédentes étaient True
6      instruction_3
7  else:
8      instruction_4 # si aucune condition précédente n'est vraie alors l'instruction_4 est exécutée

```

Mission 3.1.

Ecrire une version optimisée du programme précédent

(qui demande à l'utilisateur deux nombres entiers et affiche des messages après comparaison)

Mission 3.2.

Ecrire un programme qui demande à l'utilisateur un nombre entier et affiche s'il est pair ou s'il est impair.

Mission 3.3.

Programmer un détecteur de mensonge.

Poser une série de question à l'utilisateur et détecter s'il ment ou s'il dit la vérité.

NB: on peut poser la même question mais en l'espaçant dans le temps, ou poser la question inverse, ou poser des question permettant des recollements,...

Vidéo