

Python : les conditions:

1. les expressions et les booléens:

Si quelqu'un vous dit que "4 est égal à 5", vous lui répondez quoi ? "c'est faux". Si maintenant la même personne vous dit que "7 est égal à 7", vous lui répondrez bien évidemment que "c'est vrai".

En Python, ces deux affirmations ("4 est égal à 5" et "7 est égal à 7") s'écriront "4 == 5" et "7 == 7" (notez bien le double signe égal).

Exercice 22: Dans la console, tapez :

```
0 4 == 5
```

prgm.py

En Python, "4 == 5" est appelé une expression, une expression est soit vraie ("True"), soit fausse ("False").

Pour l'instant nous avons vu deux grands types de données : les nombres (entier ou flottant) et les chaînes de caractères, il existe un troisième type tout aussi important que les deux premiers : les booléens. Un booléen est un type de données qui ne peut prendre que deux valeurs : vrai ("True") ou faux ("False"). Une expression (comme par exemple "4 == 5") est soit True, soit False.

ATTENTION : notez le double égal "==" qui permet de distinguer une expression et une affectation (référencement d'une valeur par une variable). Le fait de confondre le "simple égal" et le "double égal" est une erreur classique qu'il faut éviter.

Exercice 23: Soit le programme suivant :

```
0 a = 4  
  b = 7
```

prgm.py

Quel est le résultat attendu après l'exécution de ce programme si vous saisissez dans la console "a==b" ?

Exercice 24: Soit le programme suivant :

```
0 a = 4  
  b = 7
```

prgm.py

Quel est le résultat attendu après l'exécution de ce programme si vous saisissez dans la console "a!=b" ?

Notez aussi l'existence des opérateurs :

- "strictement inférieur à" <
- "strictement supérieur à" >
- "inférieur ou égal à" <=
- "supérieur ou égal à" >=

À chaque fois ces opérateurs sont True (vrai) ou False (faux).

Exercice 25: Soit le programme suivant :

```
0 a = 4
  b = 7
```

prgm.py

Quel est le résultat attendu après l'exécution de ce programme si vous saisissez dans la console "a < b" ?

2. Les conditions:

Nous allons maintenant étudier une structure fondamentale en programmation le "si ...alors...sinon...".

L'idée de base est la suivante :

```
0 if expression:
  instruction_1
2 sinon:
  instruction_2
```

prgm.py

Si "expression" est True alors "suite_instruction1" est exécuté et "suite_instruction2" est ignoré.

Sinon (sous-entendu que "expression" est False) "suite_instruction2" est exécuté et "suite_instruction1" est ignoré.

Notez l'indentation suite_instruction1 et de suite_instruction2

Exercice 26: Soit le programme suivant :

```
0 a = 4
  b = 7
2 if a<b:
  print('a est plus petit que b')
4 else:
  print('a est plus grand que b')
```

prgm.py

Quel est le résultat attendu après l'exécution de ce programme ?

Vérifiez votre hypothèse en testant le programme.

Exercice 27: Écrire une fonction qui prend en paramètre un age. Si age est supérieur ou égal à 18 ans, la fonction devra renvoyer la chaîne de caractères "Bonjour, vous êtes majeur.". Si age est inférieur à 18 ans, la fonction devra renvoyer "Bonjour, tu es mineur."

Exercice 28: Soit le programme suivant :

```
0 def annonce(num, prov, dest):
    if dest != '0':
2         msg = f'le train numero {num} en provenance de {prov} et a destination de
          {dest} entre en gare'
    else:
4         msg = f'le train numero {num} en provenance de {prov} entre en gare'
    return msg
```

prgm.py

Quel est le résultat attendu après l'exécution de ce programme si vous saisissez dans la console "annonce("4557", "Paris", "Marseille")" ? Et si vous saisissez dans la console "annonce("5768", "Bonneville", "0")" ? Vérifiez votre réponse en testant ce programme.

Exercice 29: Vous êtes gérant d'un magasin et vous désirez écrire un programme Python qui calculera automatiquement le montant de la facture des clients. Tout client qui achète au moins 5 fois le même article se voit octroyer une remise de 5 % (uniquement sur le montant de l'achat de cet article). Afin de simplifier le problème, on considère qu'un client n'achète qu'un seul type d'article. Écrire une fonction qui prend en paramètre le prix unitaire de l'article et le nombre d'articles achetés. Cette fonction doit renvoyer le montant de la facture.

3. Le "ou" et le "et":

Un if peut contenir plusieurs conditions, nous aurons alors une structure de la forme :

```
0 si exp_1 op_logique exp_2:
    instruction_1
2 sinon:
    instruction_2
```

prgm.py

op_logique étant un opérateur logique.

Nous allons étudier 2 opérateurs logiques : le "ou" (noté en Python "or") et le "et" (noté en Python "and").

Par exemple (expression1 or expression2) est vrai si expression1 est vraie et expression2 est vraie.

Autre exemple (expression1 and expression2) est faux si expression1 est vraie et expression2 est faux.

Les résultats peuvent être regroupés dans ce que l'on appelle une table de vérité :

table de vérité pour le "ou"

expression1	expression2	expression1 or expression2
vrai	vrai	vrai
vrai	faux	vrai
faux	vrai	vrai
faux	faux	faux

table de vérité pour le "et"

expression1	expression2	expression1 and expression2
vrai	vrai	vrai
vrai	faux	faux
faux	vrai	faux
faux	faux	faux

Exercice 30: Soit le programme suivant :

```
0 a = 5
  b = 10
2 if a == 5 and b == 11:
    print('OK1')
4 if a == 5 or b == 11:
    print('OK2')
```

prgm.py

Quel est le résultat attendu après l'exécution de ce programme ? Vérifiez votre réponse en testant ce programme.