PYTHON 04 : les boucles

I./ Boucles inconditionnelles :

* Tapez l'exemple suivant dans l'éditeur Thonny. Comme dans l'activité précédente, attention à l'indentation du programme Python (c'est-à-dire le décalage des lignes par rapport au bord gauche de l'écran).

```
nombre = int(input("Veuillez entrer un nombre entier :"))
for i in range(nombre):
    print("L'index de la boucle est égal à", i)
print("La boucle vient de se terminer")
```

* Lancez le programme en cliquant sur la flèche verte en haut de la fenêtre Thonny. Le programme demande un nombre entier. Entrez un nombre positif de votre choix. Que se passe-t-il ? Relancez le programme puis entrez un autre nombre. Que se passe-t-il ? Que se produit-il si on tape « 0 » ou si on tape un nombre négatif ?

Le programme réalise une boucle. La variable i est l'index de la boucle : elle est initialisée à 0 au début, et augmente de 1 à chaque tour. Lorsque i est égal à (nombre – 1), la boucle s'arrête.

Attention : lorsqu'on fait « **`for i in range(10):`** », le programme va donc compter de 0 à 9. Si on veut compter de 0 à 10, il faudra penser à utiliser « **`for i in range(11):`** ».

* Tapez ensuite le programme suivant :

```
a = int(input("Veuillez entrer un premier nombre entier :"))
b = int(input("Veuillez entrer un deuxième nombre entier, plus grand :"))
for i in range(a, b):
    print("L'index de la boucle est égal à", i)
print("La boucle vient de se terminer")
```

* Lancez le programme et tapez par exemple 5 et 18. Que se passe-t-il ? Essayez avec d'autres valeurs. Que se passe-t-il si on tape un deuxième nombre qui soit plus petit ou égal au premier ?

Le programme réalise une boucle. La variable i est l'index de la boucle : elle est initialisée à « a » au début, et augmente de 1 à chaque tour. Lorsque i est égal à (b – 1), la boucle s'arrête.

* Tapez ensuite le programme suivant :

```
a = int(input("Veuillez entrer un premier nombre entier :"))
b = int(input("Veuillez entrer un deuxième nombre entier, plus grand :"))
c = 2
for i in range(a, b, c):
    print("L'index de la boucle est égal à", i)
print("La boucle vient de se terminer")
```

* Lancez le programme et tapez par exemple 5 et 18. Que se passe-t-il ? Quelle est la différence par rapport au programme précédent ? Changez la valeur de c à la troisième ligne, puis relancez le programme. Que remarque-t-on ?

Le programme réalise une boucle. La variable i est l'index de la boucle : elle est initialisée à « a » au début, et augmente de c à chaque tour. On dit que « c » est le pas de la boucle. Lorsque i est égal à (b – 1), la boucle s'arrête.

* Essayez ensuite le programme suivant :

```
for i in range(5, 0, -1):
    print("L'index de la boucle est égal à", i)
print("La boucle vient de se terminer")
```

On peut donc utiliser un pas négatif pour faire un compte à rebours.

II./ Boucles conditionnelles :

* Tapez les deux programmes suivants et testez-les :

```
nombre = 0
while nombre != 55:
    nombre = int(input("Veuillez taper 55 si vous voulez sortir de la boucle..."))
print("Nous sommes enfin sortis de la boucle !")
```

```
x = 1
while x < 10:
    print("x a pour valeur", x)
    x = x * 2
print("Fin")
```

Dans les deux cas, on s'aperçoit que la boucle va se dérouler tant qu'une condition est vraie. Dès que la condition cesse d'être vraie, on sort de la boucle.

* Tapez les deux programmes suivants et testez les :

```
for i in range(4):
    print("i a pour valeur", i)
```

```
i = 0
while i < 4:
    print("i a pour valeur", i)
    i = i + 1
```

On s'aperçoit que les deux programmes font la même chose. Cela veut donc dire qu'on peut toujours remplacer une boucle « for » par une boucle « while ». Attention, l'inverse n'est pas vrai !

En général, si on connaît avant de démarrer la boucle le nombre de tours à exécuter, on choisit une boucle « for ». Au contraire, si la décision d'arrêter la boucle ne peut se faire que par un test, on choisit une boucle « while ».

III./ Ruptures de séquences :

* Tapez les deux programmes suivants et testez-les :

```
for x in range(0, 10):
    if x == 5:
        break
    print(x)
print("Fin")
```

Le programme précédent affiche : 0, 1, 2, 3, 4, Fin : « break » permet donc de sortir de la boucle avant la fin.

```
for x in range(0, 10):
    if x == 5:
        continue
    print(x)
print("Fin")
```

Le programme précédent affiche : 0, 1, 2, 3, 4, 6, 7, 8, 9, Fin : « continue » permet donc de sauter une valeur de i.

Les deux instructions précédentes permettant de modifier la séquence de la boucle, elles sont donc appelées ruptures de séquences. On peut utiliser ces deux instructions aussi bien dans une boucle while que dans une boucle for.

Mettons maintenant en pratique ce que nous venons de voir. Pour cela vous allez devoir réaliser les programmes suivants. Pensez à les enregistrer, soit sur votre lecteur réseau, soit sur votre clef USB.

1./ Réalisez un programme qui réalise le compte à rebours d'une bombe à l'aide d'une boucle. Le programme demande un nombre entier, puis compte à rebours depuis ce nombre jusqu'à zéro, avant d'afficher « BOOOM !!! ». Par exemple, si on tape 10, on doit obtenir dans la console l'affichage suivant :

```
10
9
8
7
6
5
4
3
2
1
0
BOOOM !!!
```

2./ Réalisez un programme qui demande un nombre entier compris entre 0 et 10 inclus, puis qui affiche la table d'addition de ce nombre. Par exemple, si on tape 5, on doit obtenir :

```
0 + 5 = 5
1 + 5 = 6
2 + 5 = 7
3 + 5 = 8
4 + 5 = 9
5 + 5 = 10
6 + 5 = 11
7 + 5 = 12
8 + 5 = 13
9 + 5 = 14
10 + 5 = 15
```

3./ Réalisez un programme qui demande un nombre entier compris entre 0 et 10 inclus, puis qui affiche la table de multiplication de ce nombre. Par exemple, si on tape 7 on doit obtenir :

```
0 * 7 = 0
1 * 7 = 7
2 * 7 = 14
3 * 7 = 21
4 * 7 = 28
5 * 7 = 35
6 * 7 = 42
7 * 7 = 49
8 * 7 = 56
9 * 7 = 63
10 * 7 = 70
```

4./ Modifier le programme précédent pour qu'il affiche un message d'erreur si le nombre n'est pas compris entre 0 et 10 inclus. Il faudra certainement utiliser un test « if ».

5./ Modifier le programme précédent afin qu'il affiche un message d'erreur si le nombre n'est pas compris entre 0 et 10 inclus, et qui repose la question jusqu'à ce que l'utilisateur entre un nombre valide.

Document en ligne : https://tinyurl.com/ye8p3w87