La boucle for avec indice

La condition `if` permet d'effectuer un branchement en choisissant d'exécuter une certaine partie du code ou non. Une boucle `for` consiste à exécuter plusieurs fois une même partie du code. Cette partie, que nous appelons le corps de boucle, correspond aux lignes qui suivent l'instruction `for`. Le corps de boucle doit avoir une indentation plus importante que l'instruction `for` associée. Lorsque l'indentation revient au même niveau que celui de l'instruction `for`, cela signifie que le corps de boucle s'est achevé. La boucle `for` avec indice nécessite de spécifier le nom de la variable stockant l'indice. Juste après le mot-clé `in`, on précise la plage de valeurs que va parcourir l'indice. Par exemple, la fonction native `range(4)` définit une plage de valeurs allant de 0 à 3 : 0, 1, 2 et 3. En résumé, nous identifions le corps de boucle ainsi :

```
……
for i in range(4) :        # Démarrage boucle for - indice : i
   ……                      #  || Corps de boucle
   ……                      #  ||
   ……                      #  ||
   ……                      #  \/ Fin du corps de boucle
……
```

Si on choisit comme plage de valeurs : 0, 1, 2, lors de la première exécution du corps de boucle, la valeur de l'indice vaut 0. Lorsque le corps de boucle se termine, l'indice prend la valeur suivante, c'est-à-dire 1, et alors on exécute à nouveau le corps de boucle. On recommence, encore et encore, tant que l'indice a des valeurs à parcourir.

Lorsqu'il n'en a plus, le corps de boucle n'est plus exécuté et on passe à la ligne suivante ayant la même indentation que l'instruction `for`. Le programme poursuit alors son cours, la boucle est terminée. Mettons nos connaissances en application sur l'exemple suivant :

```
print("Avant l'instruction For")
for i in range(3) :                      # Mise en place de la boucle
  print("  Début du corps de boucle") # || Corps
  print("  i =",i)                    # ||
  print("  Fin du corps de boucle")   # \/
print("Après l'instruction For ")
```

Essayez de deviner les affichages produits par ce code. Il n'y a normalement pas de piège, l'indice de boucle `i` va démarrer à 0 et la dernière valeur qu'il prendra sera 2. Voici le résultat obtenu :

```
>> Avant l'instruction For
>>   Début du corps de boucle
>>   i = 0
>>   Fin du corps de boucle
>>   Début du corps de boucle
>>   i = 1
>>   Fin du corps de boucle
>>   Début du corps de boucle
>>   i = 2
>>   Fin du corps de boucle
>> Après l'instruction For
```

© Editions ENI – Tous droits réservés – Copie personnelle de Rafik Lachaal

Si vous êtes un peu perdu, pour vous aider, il est possible de retirer toute boucle `for` dans un programme. Le programme équivalent consiste à copier-coller le corps de boucle autant de fois que l'indice prend de valeurs. Juste avant chaque duplication du corps de boucle, on initialise l'indice à la bonne valeur. Voici le code équivalent à la boucle précédente :

```
print("Avant l'instruction For")

i = 0
print("  Début du corps de boucle")
print("  i =",i)
print("  Fin du corps de boucle")

i = 1
print("  Début du corps de boucle")
print("  i =",i)
print("  Fin du corps de boucle")

i = 2
print("  Début du corps de boucle")
print("  i =",i)
print("  Fin du corps de boucle")

print("Après l'instruction For ")
```

Pour assimiler le fonctionnement de la boucle `for`, nous vous proposons un autre exemple. Déterminez l'affichage produit par le code suivant :

```
for i in range(3) :
   print(i)
   if i == 1 :
      print("i vaut 1")
        print("----")
```

Nous allons parcourir cette boucle pas à pas :

- Démarrage de la boucle `for`
  - La fonction `range()` initialise la plage de valeurs pour l'indice `i` : 0,1 et 2
- Exécution du corps de boucle avec `i` = 0
  - L'instruction `print(i)` affiche la valeur 0
  - La condition après le `if` est fausse, on n'exécute pas le branchement
  - La fonction `print()` affiche "----"
  - Fin du corps de boucle
- Reprise de la boucle `for`, la valeur de l'indice `i` passe à 1
  - L'instruction `print(i)` affiche la valeur 0
  - La condition est vraie, on exécute le branchement sous le `if`

© Editions ENI – Tous droits réservés – Copie personnelle de Rafik Lachaal

        - La fonction `print()` affiche "i vaut 1"
        - Fin du branchement de la condition `if`
        - On passe à la ligne suivant la fin du branchement
    - La fonction `print()` affiche "----"
    - Fin du corps de boucle
- Reprise de la boucle `for`, la valeur de l'indice `i` passe à 2
    - L'instruction `print(i)` affiche la valeur 2
    - La condition après le `if` est fausse, on n'exécute pas le branchement
    - La fonction `print()` affiche "----"
    - Fin du corps de boucle
- Reprise de la boucle `for`, toutes les valeurs ont été utilisées, la boucle est terminée

Ainsi, l'affichage obtenu est le suivant :

```
>> 0
>> ----
>> 1
>> i vaut 1
>> ----
>> 2
>> ----
```

© Editions ENI – Tous droits réservés – Copie personnelle de Rafik Lachaal