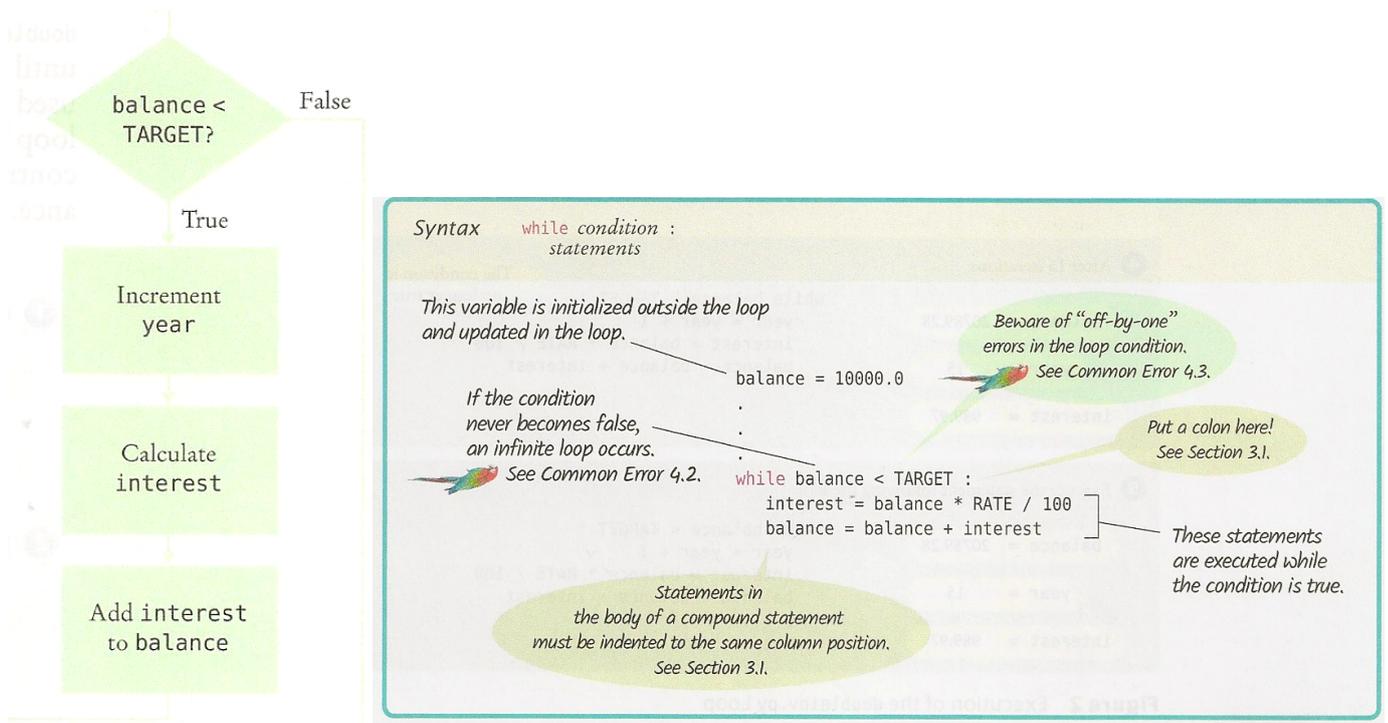


Informatique Pour Tous

Cours 4 – Boucle WHILE



A / C'est quoi une boucle `while` ? Exemple du taux d'intérêt

Vous avez placé 10000 euros sur votre compte bancaire et cela vous rapporte 5 % par an (taux très intéressant). Combien d'années vous faudra-il laisser votre argent sur votre compte pour que ce dernier atteigne la somme de 20000 euros ?

Le programme suivant répond à la question.

```
from math import *
import numpy as np
import matplotlib.pyplot as plt

#déclaration des variables
taux=5.0
somme_initiale=10000
somme_finale=20000

#initialisation des variables utilisées dans la boucle
somme=somme_initiale
annee=0

#comptage du nombre d'annees pour doubler la somme avec while
while somme<somme_finale:
    annee=annee+1
    interet=somme*taux/100
    somme=somme+interet

#affichage du resultat
print()
print("votre investissement initial a doublé après",annee,"années.")
```

votre investissement initiale a doublé après 15 années.

Comme nous avons pu le voir, une boucle est un système d'instruction permettant de répéter un certain nombre de fois toute une série d'opérations. Le mot `while` signifie "tant que" en anglais. L'instruction `while` du code ci-dessus indique à Python qu'il faut répéter continuellement le bloc d'instruction qui suit, tant que le contenu de la variable `somme` reste inférieur à 20000.

Remarques:

- 1- La variable évaluée dans la condition doit exister au préalable (affectation d'une valeur), ici la variable `somme`.
- 2- Si la condition est fausse au départ, par exemple si la somme initiale vaut 30000 euros, le corps de la boucle n'est jamais exécuté.
- 3- Si la condition reste toujours vraie, alors le corps de la boucle est répété indéfiniment ! C'est le cas ici si l'on oublie de mettre à jour la variable `somme` qui contrôle la boucle.

```
#comptage du nombre d'annees pour doubler la somme avec while
while somme<somme_finale:
    annee=annee+1
    interet=somme*taux/100
    #somme=somme+interet
```

B / Exemples instructifs simples

Exemple 1

```
i=0
total=0
while total<10:
    i=i+1
    total=total+i
    print(i,total)
```

```
1 1
2 3
3 6
4 10
```

Quand la variable `total` atteint la valeur 10, la condition sur la boucle devient fausse et la boucle prend fin.

Exemple 2

```
i=0
total=0
while total<10:
    i=i+1
    total=total-i
    print(i,total)
```

```
1 -1
2 -3
3 -6
4 -10
5 -15
6 -21
7 -28
8 -36
9 -45
10 -55
....
```

Parce que la variable `total` n'attendra jamais la valeur 10, on a une boucle infinie !

Exemple 3

```
i=0
total=0
while total<0:
    i=i+1
    total=total-i
    print(i,total)
```

La déclaration `total<0` est fausse quand la condition `while` est rencontrée pour la première fois. Ainsi la boucle n'est jamais exécutée.

Exemple 4

```
i=0
total=0
while total>=10:
    i=i+1
    total=total+i
    print(i,total)
```

Le programmeur pensait probablement « stop quand `total` est au moins égale à 10 ». Cependant, la condition sur la boucle est testée à chaque itération et pas uniquement sur la valeur finale de `total`. La déclaration `total>=0` est encore une fois fausse quand la condition `while` est rencontrée pour la première fois. Ainsi la boucle n'est jamais exécutée.

Exemple 5

```
i=0
total=0
while total>=0:
    i=i+1
    total=total+i
    print(i,total)
```

Parce que `total` est toujours plus grande ou égale à 0, la boucle tourne indéfiniment. Le programme n'affiche rien parce que la fonction `print` est en dehors du corps de la boucle à cause de l'indentation.