

Python

Processus répétitif – Boucle While

Une boucle est une structure qui permet de répéter une partie d'un programme un certain nombre de fois. La répétition se poursuit tant que la condition prévue pour la boucle demeure vraie. Lorsque la condition devient fausse, la boucle se termine et le contrôle du programme est transmis à l'instruction qui suit la boucle.

Processus répétitif – Instruction While

La boucle **while** (tant que) est l'une des structures en boucle disponibles dans Python. Elle se poursuit jusqu'à ce que l'expression devienne fausse. L'expression doit être logique et doit avoir une valeur de retour **vraie** (True) ou **fausse** (False).

Voici la syntaxe de la boucle While :

L'expression While :
Instructions

L'expression de l'expression est évaluée en premier. Si l'expression est vraie, le bloc d'instructions est exécuté de façon répétée jusqu'à ce que l'expression devienne fausse. Sinon, l'instruction suivante placée après le bloc d'instructions est exécuté.

Exemple :

```
Nombre = 0
while nombre < 9:
    print('le nombre est :', nombre)
    Nombre = nombre + 1
print ("Au revoir!")
```

Ceci produira le résultat suivant :

```
Le nombre est : 0
Le nombre est : 1
Le nombre est : 2
Le nombre est : 3
Le nombre est : 4
Le nombre est : 5
Le nombre est : 6
Le nombre est : 7
Le nombre est : 8
Au revoir!
```

Dans l'exemple ci-dessus, l'instruction finale print n'est pas indentée, ce qui indique qu'elle est en dehors de la boucle et qu'elle sera exécutée une fois que la boucle se termine. Chaque fois, au moyen de la boucle, le nombre de variables, qui est initialisé ou dont la valeur est fixée initialement à 0 avant la boucle, est incrémenté de 1. La boucle vérifie chaque fois que le nombre est toujours inférieur à 9 et, si c'est le cas, l'instruction se répète de nouveau dans la boucle.

Utilisation d'une boucle While pour exécuter de nouveau le code dans un programme

```
à partir de import.math.pi.

print ("calculatrice de volume")
print ("-----")
print ("")

# Initialiser de nouveau la variable
de nouveau = "Y"

# créer une boucle pour continuer à exécuter l'instruction tant que l'utilisateur répond
par un Y
while de nouveau == "Y":
    # demander le rayon et calculer le volume
    r = int(input("Quel est le rayon de la planète en Km"))
    v = 4/3 * pi * r ** 3

    print("Le volume de la planète est ", v)

    #demander à l'utilisateur s'il veut initialiser de nouveau la variable. La fonction
upper() convertit tous les caractères minuscules d'une réponse en majuscules.
    de nouveau = input("Souhaitez-vous calculer un autre volume? (O/N) ").upper()
```

Exercice – Élaborez un programme qui utilise une boucle while pour calculer la force gravitationnelle entre deux éléments compte tenu de leurs masses m_1 et m_2 en kg, de la distance r qui les sépare l'un de l'autre en mètres et de la constante gravitationnelle $G = 6,67 * 10^{-11}$. La boucle doit se poursuivre autant de fois que l'utilisateur le souhaite en utilisant une variable comme la variable "de nouveau".

Deux éléments, même vous et votre ami, auront une attraction gravitationnelle l'un sur l'autre. L'attraction gravitationnelle entre vous et votre ami sera négligeable par rapport à celle de la Terre ou du Soleil, mais elle existe. La formule utilisée pour calculer la force gravitationnelle est la suivante :

$F_g = (G * m_1 * m_2)/r^2$ et les unités finales sont N pour Newtons, l'unité de mesure utilisée pour mesurer toutes les forces. Affichez votre réponse par la suite pour dire :

La force entre les deux éléments est [réponse] N.

N'oubliez pas de demander à l'utilisateur s'il souhaite répéter le code pendant qu'il est encore dans la boucle.

Une fois que vous aurez une boucle de travail, vous pourrez voir la valeur de la programmation. Cela peut prendre un peu plus de travail au départ, mais une fois que c'est fait, vous pouvez effectuer des calculs infinis en une fraction de seconde.