17_python_while_loop

November 27, 2023

1 Python While loop

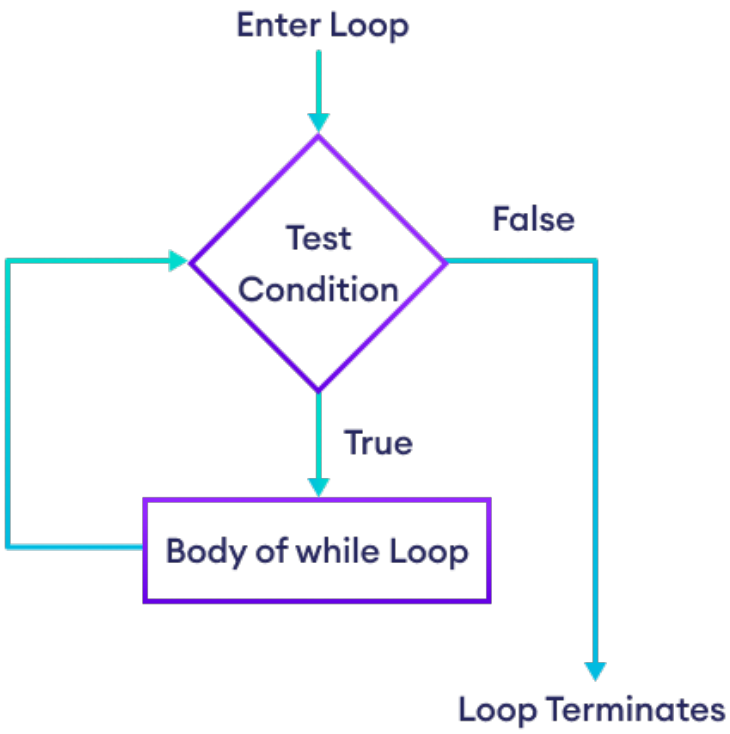
En programmation, la boucle `while` est une structure de contrôle qui permet d'exécuter un bloc d'instructions de manière répétée tant qu'une condition est vraie (`True`).

La syntaxe de la boucle `while` est la suivante:

```
while condition:
    # instructions
```

1. La boucle `while` evalue une condition
2. Si la condition est vraie, le bloc d'instructions (le body de la boucle `while`) est exécuté
3. La condition est à nouveau évaluée
4. Si la condition est toujours vraie, le bloc d'instructions est exécuté à nouveau, jusqu'à ce que la condition soit fausse
5. Lorsque la condition est fausse, le programme sort de la boucle `while` et continue à s'exécuter après la boucle `while`

2 Flowchart de la boucle `while`



Source

2.0.1 Exemple 1: Boucle `while`

[ ]:
```
# initialize the variable
i = 1
n = 5

while i <= n:
    print(i)
    i += 1
```

2.0.2 Exemple 2: Boucle `while`

[ ]:
```
# program to calculate the sum of numbers
# until the user enters zero

total = 0

number = int(input('Enter a number: '))

# add numbers until number is zero
while number != 0:
    total += number     # total = total + number
```

```
    # take integer input again
    number = int(input('Enter a number: '))


print('total =', total)
```

3 Boucles infinies

Si la condition de la boucle `while` est toujours vraie, la boucle `while` s'exécutera pour toujours, ce qui est appelé une boucle infinie. Assuré vous que la condition de la boucle `while` devienne fausse à un moment donné ou qu'il y ait une instruction `break` pour sortir de la boucle `while`.

Une boucle infinie est une boucle qui n'a pas de condition de sortie, ce qui signifie qu'elle ne s'arrête jamais. Les boucles infinies sont très utiles dans certains cas, par exemple lorsque vous souhaitez exécuter un programme en continu, mais elles sont aussi très dangereuses car elles peuvent causer des problèmes de performance et même faire planter votre ordinateur.

[ ]:
```
age = 32

# the test condition is always True
while age > 18:
    print('You can vote')
```

4 Python while loop with else

La boucle `while` peut avoir une clause `else` qui est exécutée lorsque la condition de la boucle `while` devient fausse.

[ ]:
```
counter = 0

while counter < 3:
    print('Inside loop')
    counter = counter + 1
    if counter == 3:
        break
else:
    print('Inside else')
```

Note: La clause `else` n'est pas exécutée si vous sortez de la boucle `while` avec une instruction `break`.

[ ]:
```
counter = 0

while counter < 3:
    # loop ends because of break
    # the else part is not executed
    if counter == 1:
```

```
        break

    print('Inside loop')
    counter = counter + 1
else:
    print('Inside else')
```