



LES BOUCLES



01

DÉFINITION

Qu'est ce qu'une boucle en programmation?

02

BOUCLE FOR

Comment utiliser une boucle FOR?

03

BOUCLE WHILE

Comment utiliser une boucle WHILE?

04

BOUCLES IMBRIQUÉES

Comment utiliser des boucles imbriquées?

05


INSTRUCTIONS DE CONTRÔLE

Qu'est ce que les instructions de contrôles?

06

INSTRUCTION BREAK

Comment arrêter une boucle?





07


INSTRUCTION CONTINUE

Comment passer à l'élément
suivant de la boucle?

08

INSTRUCTION PASS

Pourquoi exécuter une
déclaration nulle?



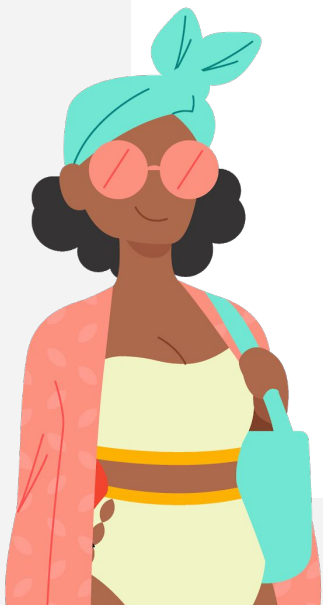
A decorative border at the top of the slide featuring a symmetrical arrangement of tropical leaves in shades of green, red, pink, and yellow. The leaves include monstera and palm-like varieties.

DÉFINITION

Les **boucles** sont des **structures** très utiles en programmation. Il vous arrivera sûrement de vouloir **exécuter une instruction** ou un **groupe d'instructions** plusieurs fois, les boucles sont faites pour cela.

En général, les instructions sont exécutées de manière **séquentielle** : La première est exécutée en premier, suivie de la deuxième, et ainsi de suite.

Python fournit **différents types de boucles** pour répondre à vos besoins, ainsi que des **structures de contrôle** qui permettent des chemins d'exécution plus compliqués.

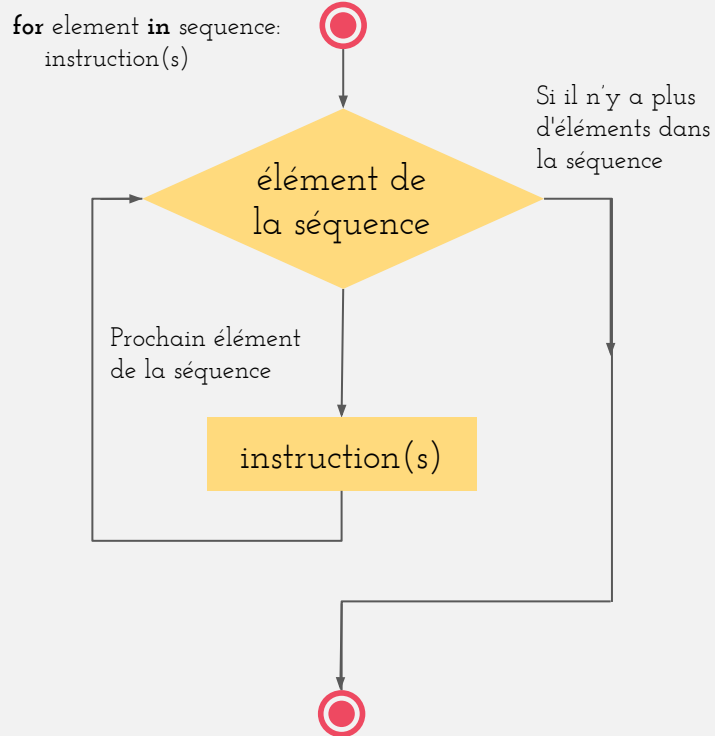


BOUCLE FOR

La boucle **FOR** a la capacité d'itérer sur les éléments de n'importe quelle séquence, comme une **liste** ou une **chaîne de caractères**.

Si une séquence contient une **liste d'expressions**, elle est évaluée en premier lieu. Ensuite, le premier élément de la séquence est attribué à la **variable itérative**. Enfin, le bloc d'instructions est **exécuté**.

Chaque élément de la liste est affecté à la **variable itérative**, et le bloc d'instructions est exécuté jusqu'à ce que la séquence entière soit épuisée.





BOUCLE FOR

Dans le premier cas la boucle va itérer sur une **chaîne de caractères** et une **liste** dans le second cas. L'instruction d'affichage va donc être **appelée à plusieurs reprises** jusqu'à **épuisement des éléments** de la séquence.

À chaque **itération**, la **valeur de l'élément courant** de la **séquence** est **affiché** avant de passer à l'élément suivant jusqu'à **épuisement des éléments** contenus dans la **séquence**.

```
for letter in 'Python':  
    print('Current Letter :', letter)
```

```
fruits = ['banana', 'apple', 'mango']  
for fruit in fruits:  
    print('Current fruit :', fruit)
```

Nous obtenons le résultat suivant :

```
Current Letter : P  
Current Letter : y  
Current Letter : t  
Current Letter : h  
Current Letter : o  
Current Letter : n  
Current fruit : banana  
Current fruit : apple  
Current fruit : mango
```

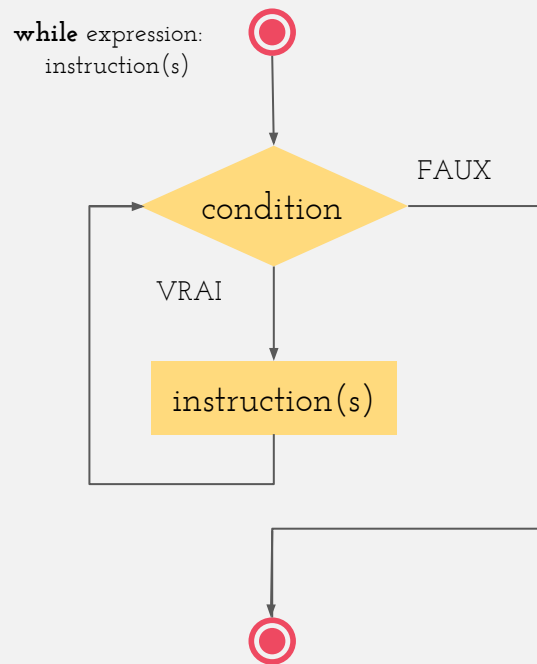
BOUCLE WHILE

La boucle **WHILE** répète une instruction tant que la **condition** donnée est **VRAIE**. La **condition** peut être n'importe quelle **expression**.

La boucle itère tant que la **condition** est **VRAIE**.

Lorsque la **condition** devient **FAUSSE**, le programme passe à la ligne qui suit immédiatement la boucle.

ATTENTION, si votre **condition** est toujours **VRAIE** votre programme effectuera à l'infini (*ou jusqu'à saturation de la mémoire de votre ordinateur*) les instructions contenues dans la boucle. Dans le cas où la **condition** est **FAUSSE** à la première itération, les instructions dans la boucle ne seront pas exécutées.





BOUCLE WHILE

Dans cet exemple, le bloc de code contenu dans la structure **WHILE** est composé des instructions **d'affichage** et **d'incrémentation**. Il est exécuté à plusieurs reprises jusqu'à ce que le décompte ne soit plus inférieur à 9.

À chaque itération, la valeur actuelle du **compteur** de l'index est **affichée**, puis **augmentée** de 1.

```
count = 0
while (count < 9):
    print('The count is:', count)
    count += 1 # count = count + 1
```

Nous obtenons le résultat suivant :

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
The count is: 6
The count is: 7
The count is: 8
```




BOUCLES IMBRIQUÉES

Python permet d'utiliser n'importe quels types de boucle à l'intérieur d'une autre boucle. Dans l'exemple ci-dessous nous avons deux boucles **WHILE** imbriquées pour trouver tous les nombres premiers jusqu'à 50.

```
i = 2
while(i < 50):
    j = 2
    while(j <= (i / j)):
        if not(i % j): break
        j = j + 1
    if (j > i / j) : print(i, " is prime")
    i = i + 1
```

Nous obtenons le résultat suivant :

```
2 is prime
3 is prime
5 is prime
7 is prime
11 is prime
13 is prime
17 is prime
19 is prime
23 is prime
29 is prime
31 is prime
37 is prime
41 is prime
43 is prime
47 is prime
```



BOUCLES IMBRIQUÉES

Nous avons 100 cerises, donc la première condition (`cherries < 200`) est validé.

Ensuite, nous retrouvons une structure en **IF...ELIF...ELSE**. La première condition de cette série est invalidée (`cherries == 150`), la condition suivante en **ELIF** est quand à elle validée (`cherries == 100`).

```
cherries = 100
if cherries < 200:
    print("There are less than 200 cherries.")
    if cherries == 150:
        print("There are 150 cherries.")
    elif cherries == 100:
        print("There are 100 cherries.")
    elif cherries < 100:
        print("There are less than 100 cherries.")
else:
    print("There are more than 200 cherries.")
print "Good bye!"
```

Nous obtenons le résultat suivant :

```
There are 100 cherries.
Good Bye
```

A decorative border at the top of the page featuring a symmetrical arrangement of tropical leaves in shades of green, red, pink, and yellow. The leaves have various shapes, including monstera and palm-like fronds.

INSTRUCTIONS DE CONTRÔLE

Les instructions de contrôle des boucles modifient l'exécution normale de la séquence. Il en existe trois types :

CONTINUE : Renvoie au début de la boucle.

BREAK : Permet de sortir de la boucle.

PASS : Permet d'écrire des boucles vides. Il est également utilisé pour les fonctions et les classes vides.





INSTRUCTION CONTINUE

L'instruction **CONTINUE** renvoie au début de la boucle. Elle rejette toutes les instructions restantes dans l'itération de la boucle et ramène au début de la boucle.

L'instruction **CONTINUE** peut être utilisée dans les boucles **WHILE** et **FOR**.

```
for letter in 'Kiwi':  
    if letter == 'i':  
        continue  
    print('Current Letter :', letter)
```

```
var = 3  
while var > 0:  
    print('Current variable value :', var)  
    var -= 1 # équivaut à var = var - 1  
    if var == 2:  
        continue
```

Résultats :

```
Current Letter : K  
Current Letter : w  
Current variable value : 3  
Current variable value : 1  
Current variable value : 0
```



INSTRUCTION BREAK

L'instruction **BREAK** met fin à la boucle et reprend l'exécution à la prochaine instruction.

L'utilisation la plus courante de **BREAK** est le déclenchement d'une condition externe nécessitant une sortie précipitée de la boucle. Elle peut être utilisée dans les boucles **WHILE** et **FOR**.

Pour les boucles imbriquées, l'instruction **BREAK** arrête l'exécution de la boucle la plus interne et commence à exécuter la ligne de code suivante après le bloc.

```
for letter in 'Kiwi':  
    if letter == 'i':  
        break  
    print('Current Letter :', letter)
```

```
var = 10  
while var > 0:  
    print('Current variable value :', var)  
    var -= 1 # équivaut à var = var - 1  
    if var == 6:  
        break
```

Résultats :

```
Current Letter : K  
Current variable value : 10  
Current variable value : 9  
Current variable value : 8  
Current variable value : 7
```



INSTRUCTION PASS

L'instruction **PASS** est utilisée lorsqu'une déclaration est **requise de manière syntaxique** mais que vous ne voulez pas qu'une **commande** ou un **code** soit exécuté.

L'instruction **PASS** est une **opération nulle**, cela veut dire que **rien ne se passe lorsqu'elle s'exécute**. Elle est également utile dans les endroits où votre code ira éventuellement, mais n'a pas encore été écrit.

```
for letter in 'Mango':  
    if letter == 'n':  
        pass  
        print('This is pass block')  
    print('Current Letter :', letter)
```

```
print('Miam miam!')
```

Resultats

```
Current Letter : M  
Current Letter : a  
Current Letter : n  
This is pass block  
Current Letter : g  
Current Letter : o  
Miam miam!
```

The background of the slide is a vibrant, colorful pattern of various tropical leaves and foliage. The leaves are in shades of green, yellow, orange, red, and purple, creating a dense and lively border around the central text. Some leaves are large and broad, while others are thin and feathery. The overall style is modern and artistic.

EXERCICES

EXERCICES

1. Calculez la somme des nombres allant de 1 à 100 en utilisant une boucle FOR.

2. Comptez le nombre de 'e' dans le paragraphe suivant en utilisant une boucle FOR :

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc egestas dolor ut tristique mattis. Suspendisse id egestas sapien, eget dictum tellus. Donec varius quis dui id laoreet. Vestibulum scelerisque nunc eu orci lobortis congue. In pellentesque aliquam lacinia. Suspendisse pellentesque lacinia ipsum in pharetra. Mauris viverra lectus id ipsum ultrices, quis lacinia diam gravida. Pellentesque placerat diam eget nisl varius, vel lacinia ante luctus. Fusce a venenatis libero. Aliquam erat volutpat.

3. Affichez les multiples de 2, de 0 à 10 en utilisant une boucle WHILE.



EXERCICES

4. Trouvez la plus petite valeur de n telle que $3*1 + 3*2 + 3*3 + \dots + 3*n$ ne dépasse pas 96873 en utilisant une boucle WHILE.

5. Dessinez une pyramide d'étoiles d'une hauteur de 7 lignes comme ci-dessous :

```
 *
***
*****
*****
*****
*****
*****
```

6. Comptez le nombre de 'a' dans le paragraphe défini précédemment. Au delà de 10 occurrences trouvées, sortir de la boucle et afficher le nombre de 'a' trouvés.

7. Avec l'aide d'une boucle FOR, affichez le paragraphe sans les espaces.



A vibrant, colorful background featuring a dense pattern of various tropical leaves and foliage. The leaves are in shades of green, yellow, orange, red, and purple, creating a lush and energetic feel. Some leaves have detailed vein patterns, while others are solid colors. The overall composition is a border-like frame around a central white area.

SOLUTIONS

CORRECTION

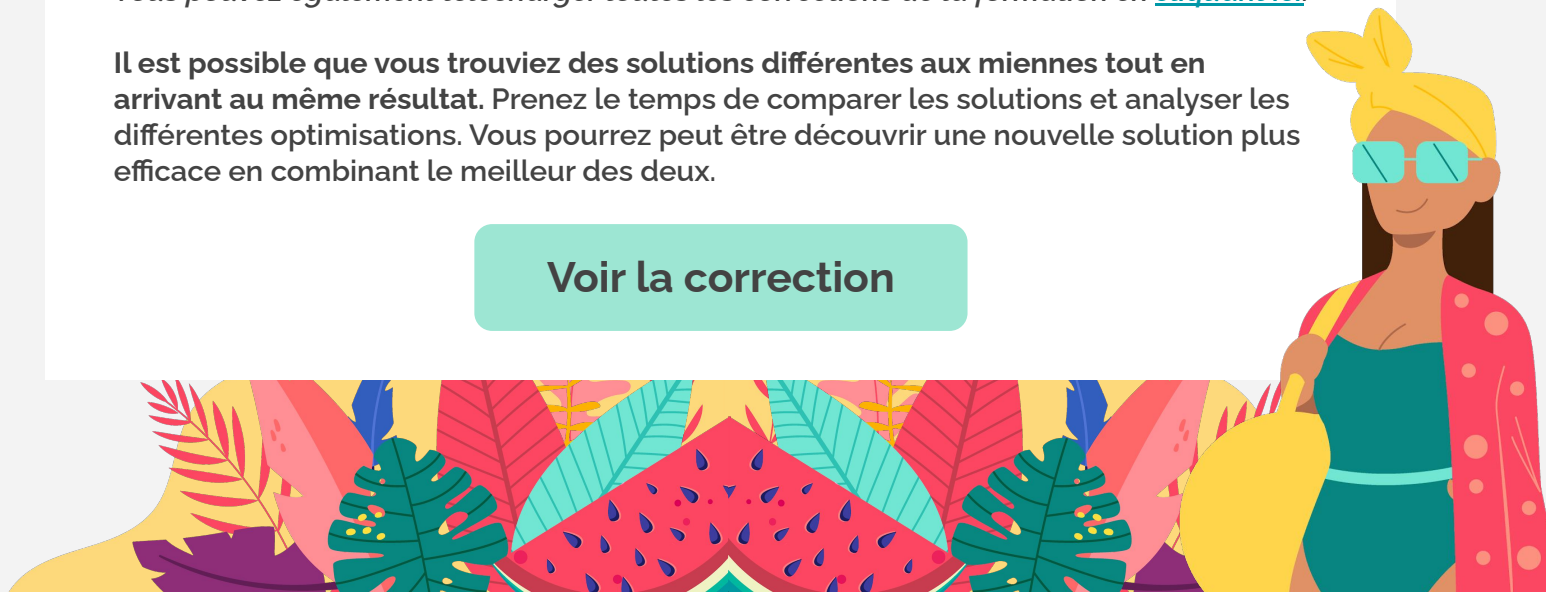
Pour visualiser la correction du chapitre cliquer sur le bouton ci-dessous.

*Le fichier de la correction s'ouvrira dans un nouvel onglet de votre navigateur préféré.
Pour cela vous devez avoir accès à une connexion Internet.*

Vous pouvez également télécharger toutes les corrections de la formation en [cliquant ici](#).

Il est possible que vous trouviez des solutions différentes aux miennes tout en arrivant au même résultat. Prenez le temps de comparer les solutions et analyser les différentes optimisations. Vous pourrez peut être découvrir une nouvelle solution plus efficace en combinant le meilleur des deux.

Voir la correction





**Félicitation vous avez terminé
le chapitre sur les boucles
avec Python!**

A decorative border of various tropical leaves in vibrant colors like red, orange, yellow, green, and purple surrounds the central white area. The leaves include Monstera, palm, and other exotic foliage.

CRÉDITS

- Modèle de la présentation par [Slidesgo](#)
- Icônes par [Flaticon](#)
- Images et infographies par [Freepik](#)