

Les boucles `while` en Python : Tout ce qu'il faut savoir

Les boucles `while` sont un outil puissant en Python pour exécuter un bloc de code tant qu'une condition donnée est vraie. Voici une explication claire et concise pour bien comprendre leur fonctionnement.

1. Syntaxe de base

```
while condition:  
    # Bloc de code à exécuter
```

- `condition` : Une expression logique (par exemple, `x < 10`) qui est évaluée avant chaque itération.
- Tant que la condition est **vraie**, le bloc de code est exécuté.
- Si la condition devient **fausse**, la boucle s'arrête.

2. Exemple simple

```
i = 1  
while i <= 5:  
    print(i)  
    i += 1
```

Résultat :

```
1  
2  
3  
4  
5
```

- Ici, la boucle s'arrête lorsque `i` dépasse 5.

3. Boucle infinie

Si la condition reste toujours vraie, la boucle ne s'arrête jamais. Par exemple :

```
while True:
    print("Ceci est une boucle infinie !")
```

Pour éviter cela, assurez-vous que la condition peut devenir fausse ou utilisez une instruction comme `break` pour sortir de la boucle.

4. Utilisation de `break` et `continue`

- `break` : Permet de sortir immédiatement de la boucle.
- `continue` : Saute à l'itération suivante sans exécuter le reste du bloc.

Exemple avec `break` :

```
i = 1
while i <= 10:
    if i == 5:
        break
    print(i)
    i += 1
```

Résultat :

```
1
2
3
4
```

Exemple avec `continue` :

```
i = 0
while i < 5:
    i += 1
    if i == 3:
        continue
    print(i)
```

Résultat :

1

2

4

5

5. Cas d'usage typiques

- **Attente d'une condition** : Par exemple, attendre une entrée utilisateur valide.

```
password = ""  
while password != "1234":  
    password = input("Entrez le mot de passe : ")  
print("Accès autorisé !")
```

- **Calculs répétitifs** : Par exemple, calculer une somme jusqu'à une condition.

```
somme = 0  
nombre = 1  
while nombre <= 10:  
    somme += nombre  
    nombre += 1  
print("La somme est :", somme)
```

6. Attention aux erreurs courantes

- **Boucles infinies accidentelles** : Si la condition ne devient jamais fausse.
- **Mauvaise gestion des variables** : Ne pas oublier de modifier les variables utilisées dans la condition.

Les boucles `while` sont idéales lorsque vous ne savez pas à l'avance combien d'itérations seront nécessaires, contrairement aux boucles `for` qui sont souvent utilisées pour un nombre fixe d'itérations.