

## C - Les boucles

Quand tu veux qu'un programme **répète plusieurs fois** une même action, tu utilises une **boucle**. C'est un mécanisme essentiel en programmation : il permet d'automatiser des tâches répétitives sans devoir copier-coller le même code plusieurs fois.

5TTR

6TTR

 Découverte

Quand tu veux qu'un programme **répète plusieurs fois** une même action, tu utilises une **boucle**. C'est un mécanisme essentiel en programmation : il permet d'automatiser des tâches répétitives sans devoir copier-coller le même code plusieurs fois.

Par exemple, si tu veux afficher les nombres de 1 à 10, tu pourrais écrire dix fois `printf()`. Mais une **boucle** permet de le faire en **trois lignes** : le programme recommence automatiquement tant qu'une condition est vraie.

## Pourquoi utiliser une boucle ?

Une boucle sert à :

- **répéter une instruction** ou un bloc d'instructions ;
- **éviter les répétitions manuelles** de code ;
- **rendre le programme plus compact et plus clair** ;

Les boucles sont donc **indispensables** pour traiter des données, parcourir des tableaux, ou effectuer des calculs successifs.

## Les trois grands types de boucles en C

1. `while` – "tant que" → On répète une action **tant qu'une condition est vraie**. Exemple : lire des nombres tant qu'ils sont positifs.
2. `do ... while` – "faire ... tant que" → Même principe, mais la condition est testée **à la fin**. Cela garantit **au moins une exécution** du bloc.
3. `for` – "pour" → La boucle la plus compacte pour répéter un nombre précis de fois. Exemple : afficher les nombres de 1 à 10.

# Exemple simple

---

```
1 | int i = 1;
2 | while (i <= 5) {
3 |     printf("Tour %d\n", i);
4 |     i++;
5 | }
```

Résultat :

Tour 1  
Tour 2  
Tour 3  
Tour 4  
Tour 5

Le programme **répète les instructions** entre `{` et `}` tant que `i <= 5`. Quand `i` devient 6, la condition est fausse, la boucle s'arrête.

---

## Bonnes pratiques

---

- Ne crée **jamais une boucle infinie involontairement** : vérifie toujours qu'une condition finit par devenir fausse.
  - Utilise des **noms de variables clairs** (comme `compteur`, `index`, `i` pour un compteur).
  - Commente ton code si la condition de la boucle n'est pas évidente.
  - Choisis le **type de boucle** adapté à la situation :
    - `while` → tu ne sais pas combien de fois il faudra répéter.
    - `for` → tu sais combien de fois tu veux répéter.
    - `do...while` → tu veux exécuter **au moins une fois**.
- 

## À retenir

---

- Une **boucle** répète un bloc d'instructions tant qu'une condition est vraie.
- Il existe **trois types principaux** : `while`, `do...while`, et `for`.
- Une boucle a toujours :
  - une **condition** (quand s'arrêter),

- un **bloc d'actions** (quoi faire),
- souvent une **variable de contrôle** (compteur ou état).

Dans les prochains articles, tu apprendras à utiliser chaque type de boucle en détail, avec des exemples concrets et des exercices.