

C - Les boucles `for`

En programmation, la boucle `for` est la plus utilisée lorsqu'on **sait d'avance combien de fois** une action doit être répétée. Elle est compacte, claire, et très pratique pour parcourir des séquences de nombres ou de positions dans un tableau.

5TTR

6TTR

 Découverte

Principe

La boucle `for` permet d'exécuter un bloc d'instructions un certain nombre de fois, en **initialisant**, **testant**, puis **modifiant** une variable de contrôle à chaque tour.

Sa syntaxe est la suivante :

```
1 | for (initialisation; condition; mise à jour) {  
2 |     // instructions à répéter  
3 | }
```

Ces trois parties se comportent comme suit :

1. **initialisation** → exécutée **une seule fois** avant le début de la boucle ;
2. **condition** → testée avant **chaque tour** (si elle devient fausse, la boucle s'arrête) ;
3. **mise à jour** → exécutée **après chaque tour**.

Exemple de base

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

Résultat :

```
Tour 1  
Tour 2  
Tour 3  
Tour 4  
Tour 5
```

Décryptage

- `int i = 1` : on crée un compteur `i` qui démarre à 1 ;
- `i <= 5` : la boucle continue **tant que** `i` est inférieur ou égal à 5 ;
- `i++` : à la fin de chaque tour, on **incrémente** `i` de 1.

Quand `i` devient 6, la condition est fausse et la boucle s'arrête.

Exemple concret

```
1 | int somme = 0;
2 | for (int i = 1; i <= 10; i++) {
3 |     somme += i; // ajoute i à la somme
4 | }
5 | printf("Somme = %d\n", somme);
```

Ce programme calcule la somme des nombres de 1 à 10.

Variante : décrémentation

Tu peux aussi faire une boucle qui **compte à rebours** :

```
1 | for (int i = 10; i >= 1; i--) {
2 |     printf("%d\n", i);
3 | }
4 | printf("Décollage !\n");
```

Bonnes pratiques

- Utilise `for` lorsque **le nombre d'itérations est connu** à l'avance.
- Choisis un **nom de compteur clair** (`i`, `j`, `k`) pour des boucles imbriquées, ou `compteur`, `index` pour plus de clarté).
- N'oublie **jamais la condition d'arrêt**, sinon tu crées une boucle infinie.
- Évite de modifier la variable de contrôle **dans le bloc** : garde-la uniquement dans la partie `mise à jour`.

À retenir

- `for` est idéale pour répéter une action **un nombre précis de fois**.
- Elle regroupe en une seule ligne :
 - l'initialisation,
 - le test de condition,
 - la mise à jour.
- C'est la boucle la plus pratique pour **parcourir des tableaux ou générer des suites de nombres**.

Dans la suite, tu verras les boucles `while` et `do...while`, qui servent quand le nombre de répétitions **n'est pas connu d'avance**.