



# Boucle

Une boucle permet d'exécuter plusieurs fois un bloc de code, tant que la condition est vraie.





# Boucle while

La boucle while permet de répéter un bloc de code tant qu'une condition est vraie.

```
while (condition) // tant que la condition est vraie  
{  
  // bloc de code à répéter  
}
```



# Exemple pratique

Les boucles `while` sont très pratiques pour, un exemple: itérer sur un tableau.

Rappel,

Un tableau possède plusieurs éléments d'un même type;

Un tableau possède une fonction qui permet de connaître sa longueur : `.Length` ;



# Exemple pratique

```
// parcourir son tableau avec un boucle while
int[] tableau = { 1, 2, 3, 4, 5 };

int index = 0;
while (index < tableau.Length) // tant que index est plus petit que la longueur du tableau
{
    Console.WriteLine(tableau[i]); // afficher l'element i du tableau
    index++; // augmenter index de 1
}
```

A partir de maintenant nous utiliserons une convention de code pour nommer une variable **index** : **i**



# Boucle for

La boucle for permet de répéter un bloc de code un nombre défini de fois.

```
for (initialisation; condition; incrémentation)
{
    // bloc de code a repeter
}
```



# Boucle for

Mots clefs :



`for`



`initialisation`



`condition`



`incrémentation`



# Boucle for

```
// parcourir son tableau avec un boucle while  
int[] tableau = { 1, 2, 3, 4, 5 };  
  
for (int i = 0; i < tableau.Length; i++)  
{  
    Console.WriteLine(tableau[i]); // afficher l'element i du tableau  
}
```

On voit bien ici, contrairement à la boucle `while`, `i` s'incrémente automatiquement à la fin du bloc conditionnel.



# Boucle foreach

La boucle for each permet de parcourir un tableau sans avoir besoin de connaître sa longueur.

```
foreach (type element in tableau)
{
    // bloc de code a repeter
}
```





# Boucle foreach

Mots clefs :



`foreach`



`in`



# Boucle foreach

```
// parcourir son tableau avec un boucle while  
int[] tableau = { 1, 2, 3, 4, 5 };  
  
foreach (int elems in arr)  
{  
    Console.WriteLine(tableau[elems]); // afficher l'element elems du tableau  
}
```

On voit bien ici, contrairement à la boucle `while`, `elems` s'incrémente automatiquement à la fin du bloc conditionnel.



# Boucle infinie et sortie

Si la condition d'arrets de la boucle n'est jamais vraie, alors votre code bouclera à l'infinie.

On peut trouver des intérêts à créer ce genre de boucle, sachant que l'on peut sortir d'une boucle avec le mot clef :

`break`

```
while (true) {  
    string? txt = Console.ReadLine();  
    if (txt == "exit") {  
        break;  
    }  
}
```