

# Structure itérative (suite)

- do-while
- foreach
- break
- continue

## Boucle `do-while`

Très semblable à la boucle While. La boucle `do-while` exécute son bloc de code **au moins une fois**, puis continue tant que la condition est vraie.

- La condition est vérifiée **après** l'exécution du bloc de code.
- Si la condition est vraie, le bloc de code est exécuté à nouveau.

## Boucle **do-while** C#

```
int compteur = 0;

do{
    Console.WriteLine(compteur);
    compteur++;
}
while (compteur < 5);
```

## Boucle `foreach`

La boucle `foreach` permet d'itérer sur les éléments d'une collection (tableau, liste, etc.).

- `foreach (type variable in collection)` : pour chaque élément de la collection, exécute le bloc de code, avec `variable` représentant l'élément en cours.
- Pratique pour parcourir des collections sans avoir à gérer les index manuellement.

```
int[] nombres = [1, 2, 3, 4, 5];  
  
foreach (int nombre in nombres){  
    Console.WriteLine(nombre);  
}
```

# L'instruction `break`

L'instruction `break` est utilisée pour **sortir** d'une boucle avant qu'elle ne se termine normalement.

- Lorsque exécuté, `break` arrête la boucle immédiatement, même si la condition initiale de la boucle est toujours vraie.

```
while(true) {  
    string txt = Console.ReadLine();  
    Console.WriteLine(txt);  
    if (txt == "exit") {  
        break; // Sort de la boucle quand l'utilisateur écrit exit  
    }  
}
```

## L'instruction `continue`

L'instruction `continue` permet de **passer** directement à l'itération suivante, sans exécuter le reste du code de la boucle.

```
for (int i = 0; i < 10; i++) {  
    if (i % 2 == 0){  
        continue; // Passe immédiatement à la prochaine itération de la boucle  
    }  
    Console.WriteLine(i); // Affichera donc seulement les nombres impairs  
}
```

# Exemples pratiques



## do-while

```
int nombre;  
do {  
    Console.Write("Entrez un nombre positif : ");  
    nombre = int.Parse(Console.ReadLine());  
}  
while (nombre < 0);
```

## foreach avec une liste

```
List<string> fruits = ["Pomme", "Banane", "Orange"];  
  
foreach (string fruit in fruits){  
    Console.WriteLine(fruit);  
}
```

## break dans une boucle

```
for (int i = 0; i < 10; i++){  
    if (i == 7){  
        break; // Sort de la boucle lorsque i est égal à 7  
    }  
    Console.WriteLine(i);  
}
```

## continue dans une boucle

```
for (int i = 0; i < 10; i++){  
    if (i % 3 == 0){  
        continue; // Ignore les multiples de 3  
    }  
    Console.WriteLine(i);  
}
```

## Conclusion

- `do-while` : Comme une boucle while, mais exécute le bloc de code au moins une première fois.
- `foreach` : Boucle sur chaque élément d'une collection.
- `break` : Arrête la boucle immédiatement.
- `continue` : Passe immédiatement à l'itération suivante sans exécuter le reste du code.