

# LES BOUCLES

En JavaScript, tout comme dans de nombreux autres langages de programmation, nous utilisons des boucles pour exécuter un bloc de code plusieurs fois.

## La boucle for

La boucle for est la plus couramment utilisée en JavaScript. Elle est généralement utilisée lorsque vous savez combien de fois le code doit être exécuté.

Voici comment cela fonctionne :

```
for (var i = 0; i < 5; i++) {  
  console.log(i);  
}
```

Dans cet exemple, la boucle commencera par  $i = 0$ , continuera tant que  $i$  est inférieur à 5, et incrémentera  $i$  de 1 à chaque itération. Ainsi, elle affichera les nombres de 0 à 4 dans la console.



## La boucle while

La boucle while est une autre boucle couramment utilisée en JavaScript. Elle exécute un bloc de code tant qu'une condition spécifiée est vraie.

Voici un exemple :

```
var i = 0;
while (i < 5) {
  console.log(i);
  i++;
}
```

Dans cet exemple, la boucle commencera par  $i = 0$  et continuera tant que  $i$  est inférieur à 5. Ainsi, elle affichera également les nombres de 0 à 4 dans la console.



## La boucle do...while

La boucle do...while est similaire à la boucle while, mais elle exécute le bloc de code une fois avant de vérifier si la condition est vraie.

Voici un exemple :

```
var i = 0;
do {
  console.log(i);
  i++;
} while (i < 5);
```

Dans cet exemple, même si la condition était fausse dès le début (par exemple, si nous avons commencé avec  $i = 5$ ), le bloc de code serait quand même exécuté une fois.



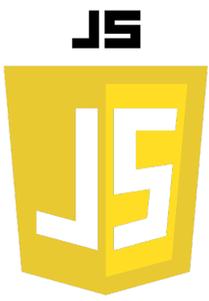
## La boucle for...in

La boucle for...in est utilisée pour parcourir les propriétés d'un objet.

Voici un exemple :

```
var person = {firstName: "John", lastName: "Doe", age: 25};  
  
for (var key in person) {  
    console.log(key + ": " + person[key]);  
}
```

Dans cet exemple, la boucle parcourt chaque propriété de l'objet person et affiche le nom de la propriété et sa valeur.



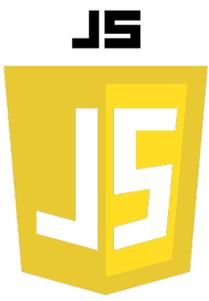
## La boucle for...of

La boucle for...of est utilisée pour parcourir les valeurs d'un objet itérable, comme un array ou une string.

Voici un exemple :

```
var colors = ["red", "green", "blue"];  
  
for (var color of colors) {  
    console.log(color);  
}
```

Dans cet exemple, la boucle parcourt chaque élément du tableau colors et affiche sa valeur.



## Contrôle de boucle

Il existe également des instructions spéciales que vous pouvez utiliser pour contrôler le flux d'une boucle :

- `break` : termine la boucle et passe à l'instruction suivant la boucle
- `continue` : termine l'itération en cours et passe à la suivante

Voici un exemple de leur utilisation :

```
for (var i = 0; i < 10; i++) {  
  if (i === 3) {  
    console.log("Skipping 3");  
    continue;  
  }  
  if (i === 7) {  
    console.log("Breaking at 7");  
    break;  
  }  
  console.log(i);  
}
```

Cet extrait de code illustre l'utilisation des instructions `break` et `continue` pour contrôler le flux d'une boucle `for`. L'instruction `continue` est utilisée pour sauter l'itération lorsque `i` est égal à 3, et l'instruction `break` est utilisée pour arrêter la boucle lorsque `i` est égal à 7.&

Ci-joint un extrait de code qui démontre l'utilisation des boucles en JavaScript. Vous pouvez le consulter et l'exécuter en suivant ce lien : <https://app.coderpad.io/sandbox?snippet=f5e5dfc1>

Cet extrait de code illustre l'utilisation des boucles `for`, `while`, `do...while`, `for...in` et `for...of` en JavaScript. Il montre également comment utiliser les instructions `break` et `continue` pour contrôler le flux d'une boucle.