

# Les boucles

[Description](#)[Sommaire](#)

## À propos de ce tutoriel

Dans ce nouveau chapitre nous allons parler des boucles. Les boucles permettent de répéter une certaine logique suivant une condition précise. Il existe plusieurs manières de créer des boucles :

## La boucle While

La boucle while permet d'exécuter un code tant que la condition passée en paramètre n'est pas satisfaite

```
while (<condition>) {  
    // ....  
}
```

Il faudra faire attention avec l'utilisation de cette boucle à ce que la condition devienne fausse à un moment donné au risque de voir la boucle se dérouler de manière infinie.

```
let i = 0  
while (i < 3) {  
    console.log("Je compte " + i)  
    i = i + 1 // peut aussi s'écrire i++  
}
```

Il est aussi possible de mettre la condition en fin de boucle grâce à la syntaxe `do...while`

```
let i = 0
do {
  console.log("Je compte " + i)
  i = i + 1
} while (i < 3)
```

Il est possible de forcer la sortie de la boucle à l'aide du mot clef `break`, ce mot clef fonctionnera avec les autres types de boucle

```
let i = 0
while (i < 3) {
  console.log("Je compte " + i)
  if (i === 1) {
    break
  }
  i++
}
```

## La boucle For

La boucle `for` permet d'exécuter un code un certain nombre de fois en précisant manuellement l'intervalle pour lequel on souhaite faire la boucle. Elle présente une notation plus concise que le `while` :

```
for (let i = 0; i < 3; i++) {
  console.log("Je compte " + i)
}
```

Il est possible de sortir de la condition à l'aide d'un `break` mais on peut aussi sauter une itération à l'aide du mot clef `continue`

```
for (let i = 0; i < 3; i++) {
  if (i === 1) {
    continue
  }
}
```

```
    console.log("Je compte " + i)
}
```

Cette boucle s'avèrera très utile lorsqu'il s'agira de parcourir un tableau

```
const eleves = ['Jean', 'Marc', 'Marie']
for (let i = 0; i < eleves.length; i++) {
    console.log(eleve[i]) // affichera alternativement : Jean, Marc, Marie
}
```

## for...in

L'instruction `for in` permet d'itérer sur les éléments énumérables. Elle permettra de récupérer les clefs d'un tableau ou les propriétés d'un objet.

```
const object = { a: 1, b: 2, c: 3 };

for (const property in object) {
    console.log(property) // affichera alternativement : a, b, c
}
```

```
const eleves = ['Jean', 'Marc', 'Marie']
for (const i in eleves) {
    console.log(i) // affichera alternativement : 0, 1, 2
}
```

## for...of

L'instruction `for of` permet de boucler sur un objet itérable en renvoyant les valeurs à chaque itération.

```
const eleves = ['Jean', 'Marc', 'Marie']
for (const eleve of eleves) {
    console.log(eleve) // affichera alternativement : Jean, Marc, Marie
}
```

# Exercices

## Décompte

08:18 - Pour cet exercice on demandera à l'utilisateur (à l'aide de prompt) un nombre à l'utilisateur entre 0 et 10, on affichera ensuite tous les nombres sous le nombre indiqué par l'utilisateur (s'il rentre 4, on affichera 4, 3, 2, 1, 0). Si l'utilisateur rentre un chiffre erroné on affichera un message d'erreur.

!!

```
let chiffre = prompt('Entrez un nombre')
if (chiffre > 10 || chiffre < 0) {
  console.log("Le nombre n'est pas entre 0 et 10")
} else {
  for (let i = chiffre; i >= 0; i--) {
    console.log(i)
  }
}
```

## Deviner le chiffre

12:53 - On créera une variable `guess` qui contiendra un nombre entre 0 et 10, l'objectif sera ensuite de faire deviner ce chiffre à l'utilisateur. Si l'utilisateur se trompe on lui donnera une indication "plus" ou "moins" pour l'orienter vers la bonne réponse.

Voir la réponse