

Explications des notions clés

Site: [Ada Tech School](#)
Cours: [GHN] [C10] Conditions
Livre: Explications des notions clés

Imprimé par: Antoine Mourin
Date: mardi 7 octobre 2025, 22:46

Table des matières

- 1. Les booléens
- 2. Les opérations logiques
- 3. Les conditions

1. Les booléens

Lors de la première itération, nous avons évoqué les types de variables dont le type booléen (boolean).

Un booléen est donc un type de variable qui peut avoir deux états en valeur :

- true ou false
- yes ou no
- on ou off

True ou false est le plus utilisé (s'il y en a un à retenir c'est celui-là !).

Les booléens sont nécessaires dans le code pour exécuter de la logique via des opérations.

Pour déclarer un booléen, c'est exactement comme une variable, sauf que sa valeur sera donc soit true, soit false. Par défaut, un booléen prend la valeur false.

```
// Je déclare un booléen en javascript qui prend l'état false
let isPair = false

# Je déclare un booléen en python qui prend l'état true
isPair = True

/* Je déclare un booléen en Java qui prend l'état true*/
boolean isPair = true;

/* Je déclare un booléen en C qui prend l'état false */
bool isPair = false;
```

2. Les opérations logiques

Les booléens sont associés aux opérations logiques.

Prenons cet exemple en JavaScript :

```
// Je déclare deux variables
let maxNumber = 10
let myNumber = 8

// Je déclare une variable qui est le résultat d'une opération logique
let myNumberBiggerThanMax = myNumber > maxNumber

console.log(myNumberBiggerThanMax)
```

- Que vois-tu dans ta console ?

Tu dois voir : false.

En effet la variable `myNumberBiggerThanMax` prend comme valeur `false` car c'est le résultat de l'opération : `myNumber > maxNumber`. Cette opération compare la valeur de `myNumber` à celle de `maxNumber` et plus exactement elle vérifie si la valeur de `myNumber` est plus grande que celle de `maxNumber`.

Il est très intéressant d'utiliser des opérations pour comparer des valeurs : savoir s'il y en a une plus grande que l'autre, si elles sont égales, si elles sont différentes. Nous pouvons aussi faire des opérations pour vérifier que des valeurs sont paires/impaires ou divisibles par un nombre par exemple. Pour réaliser toutes ces opérations, nous utilisons des opérateurs logiques.

```
let myNumber = 1252

// Je vérifie que mon chiffre est plus petit que 1000
let isMyNumberLessThan1000 = myNumber < 1000
console.log("Mon nombre est plus petit que 1000: ", isMyNumberLessThan1000)

// Je vérifie que mon chiffre est différent de 1000
let isMyNumberEqualTo1000 = myNumber !== 1000
console.log("Mon nombre est différent de 1000: ", isMyNumberEqualTo1000)

// Je vérifie que mon chiffre est plus grand ou égal à 1000
let isMyNumberBiggerOrEqualTo1000 = myNumber >= 1000
console.log("Mon nombre est égal ou plus grand que 1000: ", isMyNumberBiggerOrEqualTo1000)

// Je vérifie que mon chiffre est pair
let isPair = myNumber % 2 == 0 // le % représente le modulo
console.log("Mon chiffre est pair: ", isPair)
```

Les as-tu relevé ?

Nous avons utilisé différents opérateurs logiques pour faire des opérations de logique :

> : plus grand que

< : plus petit que

>= : plus grand que ou égal à

<= : plus petit que ou égal à

== : égal à

!= : différent de

Il en existe d'autres. Recherche ceux qui manquent à cette liste.

3. Les conditions

Nous avons compris ce qu'était un booléen et que nous pouvions faire des opérations logiques avec des opérateurs. Mais à quoi cela sert-il vraiment ?

À rajouter de la logique à notre code.

Pour faire cela, nous utilisons des blocs de conditions.

```
let myNumber = 1252
let isMyNumberLessThan1000 = myNumber < 1000

if (isMyNumberLessThan1000) {
  console.log("Mon nombre est plus petit que 1000")
} else {
  console.log("Mon nombre est plus grand ou égal à 1000")
}
```

Alors que fait notre code si nous le traduisons ? *"Si myNumber est plus petit que 1000, alors nous afficherons 'Mon nombre est plus petit que 1000' dans la console, sinon nous afficherons 'Mon nombre est plus grand ou égal à 1000'."*

Nous avons utilisé deux mots clés : **IF** / **ELSE**, qui correspondent exactement à **SI** / **SINON**.

Après le **IF**, nous avons utilisé les parenthèses pour mettre le résultat de notre opération logique. Si cette opération est vraie (**true**), alors nous passerons dans le code qui se trouve entre accolades en dessous du **IF**. Sinon, nous passerons dans le bloc de code entre accolades en dessous du **ELSE**.

Nous exécuterons donc seulement un des deux blocs de code et non pas les deux.

En effet, dans l'exemple, nous ne voulons voir qu'un seul des deux messages dans la console.

Nous avons ainsi créé notre première condition dans un code et ajouté de la logique.

Les formats de conditions

Nous avons évoqué le **IF/ELSE** et il en existe d'autres :

- **IF** / **ELSE IF** / **ELSE**
- **IF** seul
- Le **switch case**

Fais quelques recherches pour noter leur syntaxe et comment ils fonctionnent.