

# cours 2 : La modification d'une variable

Comme nous l'avons vu précédemment, en JavaScript, il existe plusieurs **types** pour décrire la valeur d'une variable. En fonction du type de données, les opérations qui permettent de manipuler ces données sont différentes.

Dans ce chapitre, nous allons apprendre à manipuler les trois types de données basiques :

- le type **string** : correspond à une chaîne de caractères, c'est-à-dire un texte ;
- le type **number** : correspond à un chiffre ;
- le type **boolean** : correspond à un booléen en mathématiques, c'est-à-dire une valeur qui est soit vraie (true, en anglais) soit fausse (false, en anglais).

## Utilisez des opérateurs

Pour modifier la valeur d'une variable, nous allons utiliser des **opérateurs**.

Un opérateur est un symbole qui permet d'effectuer des actions sur des variables ou des valeurs. Ces actions peuvent être des opérations mathématiques, ou encore des comparaisons entre valeurs.

Modifiez la valeur d'une variable avec les opérateurs :

On peut modifier les valeurs d'une variable en utilisant des opérateur par exemple :

+ addition ;

- soustraction ;

\* multiplication ;

/ division ;

= est égal à ;

Commençons par manipuler des chiffres avec une opération mathématique simple : l'addition.

Imaginons que vous souhaitiez connaître le nombre d'entrées qu'a fait le film Batman durant les deux premières semaines suivant sa sortie dans un cinéma donné. Il vous faudra additionner le nombre d'entrées de la première semaine avec celui de la deuxième semaine.

Écrivons cela en JavaScript :

```
const entreesPremiereSemaine = 1000;
```

```
const entreesDeuxiemeSemaine = 2000;
```

```
const totalEntrees = entreesPremiereSemaine + entreesDeuxiemeSemaine;
```

```
console.log(totalEntrees)
```

Ici nous avons utilisé des instructions *const*, car les valeurs ne changeront pas dans la suite du code. **N'oubliez pas de vous poser la question chaque fois que vous devez définir une variable.**

Et voilà ! Vous avez fait votre première addition en JavaScript. Le principe est le même pour les autres opérations mathématiques qui auront chacun un opérateur : les soustractions -, les multiplications \*, les divisions / .

Imaginez maintenant un autre cas de figure. Vous voulez connaître en direct le nombre de places vendues dans la journée.

Vous avez une variable qui contient le nombre actuel de places vendues, et vous voulez lui ajouter le nombre de places vendues à l'instant dans le cinéma.

Vous écrivez donc :

```
let placesDejaVendues = 150
```

```
placesDejaVendues = placesDejaVendues + 10
```

Comme vous pouvez le constater, j'ai répété deux fois *placesDejaVendues*. Si j'avais écrit *placesDejaVendues = 10*, cela aurait **écrasé** la valeur 150 au lieu de réaliser une addition. Comme ce genre d'opération est très fréquent, JavaScript a créé des raccourcis pour les réaliser : les **opérateurs d'affectation**.

```
let placesDejaVendues = 150
```

```
placesDejaVendues += 10
```

Et le résultat est strictement équivalent. 😊

Lorsqu'on utilise += , on additionne la valeur à droite du signe = à notre variable. Les variantes pour les autres opérations mathématiques existent également. On aura donc -= pour une soustraction, \*= pour une multiplication, /= pour une division.

## Manipulez des chaînes de caractères

Pour déclarer une variable qui contient une chaîne de caractères, nous devons entourer notre texte avec des guillemets simples ' ou doubles ". Regardez l'exemple ci-dessous :

```
let monTitre = "Le titre de mon article"
```

```
let monContenu = 'Le contenu de mon article'
```

Si on utilise l'opérateur + entre deux chaînes de caractères, on réalise ce que l'on appelle en programmation une **concaténation**.

La concaténation est le fait de mettre bout à bout deux chaînes de caractères. On utilise cette opération quand la chaîne de caractères que l'on souhaite stocker dans une variable est dans deux variables différentes.

Par exemple :

```
let premierePartie = "Mon nom est Bond"
```

```
let deuxiemePartie = ", James Bond."
```

```
let punchline = premierePartie + deuxiemePartie
```

```
// punchline vaut "Mon nom est Bond, James Bond."
```

Les caractères // que vous apercevez à la fin de cet extrait de code me permettent d'insérer des commentaires écrits sans qu'ils aient une incidence sur le code lui-même. C'est ainsi que je pourrai vous donner des indications directement dans le code.

Les commentaires sont d'une manière générale très utiles si vous devez vous relire par la suite, ou partager votre code avec d'autres.

Comme pour les chiffres, vous pouvez également utiliser l'opérateur += :

```
let punchline = "Mon nom est Bond"
```

```
punchline += ", James Bond."
```

```
// punchline vaut "Mon nom est Bond, James Bond."
```

## Manipulez des booléens

Très souvent, en programmation, vous aurez besoin d'un type de données qui ne peut avoir que 2 possibilités : oui/non, vrai/faux, allumé/éteint. Ainsi, une variable qui contient un booléen ne peut avoir que deux valeurs : true ou false. On utilise ce type de données pour savoir si un état est vrai ou faux dans notre code.

Par exemple, pour savoir si un utilisateur d'un site web est connecté ou pas, on écrira :

```
let connexionOk = false
```

## Différenciez les types de données

Lorsque vous utilisez des opérateurs, ne mélangez pas les types de données entre eux. Assurez-vous que vos variables ont des types cohérents.

Reprenons l'exemple des places de cinéma, mais avec des guillemets autour des chiffres.

Si on écrit ceci :

```
let placesDejaVendues = "150"
```

```
placesDejaVendues += "10"
```

```
console log(placesDejaVendues)
```

La variable *placesDejaVendues* va contenir "15010" et pas "160", car la présence des guillemets indique à JavaScript que le type de données est "String", c'est-à-dire du texte. Par conséquent, le + n'est plus un opérateur d'**addition**. Il devient un opérateur de **concaténation**, qui colle deux morceaux de texte.

Pour résoudre ce problème, nous pouvons soit enlever les guillemets autour des chiffres, soit préciser à JavaScript que nous voulons vraiment utiliser des chiffres, grâce à l'**instruction Number**. Cette dernière permet de convertir le texte "150" en chiffre 150.

```
let placesDejaVendues = Number("150")
```

```
placesDejaVendues += Number("10")
```

```
console log
```

Cette fois-ci, le résultat est correct !

Je vous propose de revenir sur cette opération dans la vidéo ci-dessous :

## En résumé

- Vous connaissez les types de données : **string**, **number**, **boolean**.
- Vous pouvez modifier une variable en utilisant des opérateurs qui seront différents en fonction du type de données que vous manipulez.
- Vous obtiendrez alors des résultats différents, comme une opération mathématique entre deux chiffres, ou une concaténation entre deux chaînes de caractères.

*Vous connaissez les différents types de données et comment les manipuler en JavaScript. Suivez-moi dans le prochain chapitre pour passer à la vitesse supérieure : structurer des données !*