

5 minutes

Objectifs du chapitre et les variables var

Objectifs du chapitre

C'est parti ! Nous allons attaquer les bases du langage JavaScript. Le cours sera toujours maintenu à jour de la dernière version des standards ECMAScript.

Nous allons commencer par aborder les variables JavaScript : var, let et const.

Nous poursuivrons en abordant le **hoisting ou remontée des variables**.

Nous continuerons avec les **types** puis avec les **opérateurs**.

Ensuite, nous verrons la **type coercion ou conversion automatique implicite**.

Puis, nous étudierons les concepts importants de **scope ou portée** et **de contexte d'exécution**.

Enfin, nous terminerons par les **références et les valeurs**.

Qu'est-ce qu'une variable ?

Une variable est un conteneur pour une valeur. Elle peut contenir toutes les valeurs JavaScript : aussi bien des chaînes de caractères, que des nombres, des tableaux, des fonctions, ou encore des objets.

Il est possible de changer la valeur contenue dans la variable, d'où leur nom.

En JavaScript, il existe trois sortes de variables : var, let et const que nous allons étudier.

Déclaration d'une variable

La première étape pour utiliser une variable est **la déclaration** :

```
var nomDeLaVariable;
```

Copier

Ici, nous disons à l'interpréteur JavaScript que nous allons avoir une variable var qui aura comme nom nomDeLaVariable. On ne sait pas encore ce qu'elle va contenir mais ce n'est pas important en JavaScript car les types sont dynamiques.

Si vous essayez d'afficher le contenu d'une variable vide mais qui a été déclarée vous aurez undefined car la valeur n'est pas définie :

```
var nomDeLaVariable;  
console.log(nomDeLaVariable); // undefined
```

Copier

En JavaScript, `undefined` est une valeur spéciale qui indique que vous n'avez assigné aucune valeur et que la valeur n'est donc pas définie.

Assignment d'une variable

Une fois que vous avez déclaré une variable, vous pouvez **l'assigner, c'est-à-dire lui donner une valeur** :

```
var nomDeLaVariable;  
nomDeLaVariable = 1;  
console.log(nomDeLaVariable); // 1
```

Copier

Vous pouvez également déclarer la variable et lui assigner une valeur immédiatement :

```
var nomDeLaVariable = 1;
```

Copier

Notez, l'utilisation d'un point virgule à la fin de chaque ligne ici. En JavaScript le point virgule indique au moteur que l'instruction est terminée et qu'il peut passer à la suivante.

En JavaScript, les points virgules ne sont pas obligatoires et le moteur les ajoutera automatiquement lorsqu'il parsera le code (lorsqu'il lira le code et l'analysera).

Cependant, **nous recommandons fortement de ne pas se reposer sur l'insertion automatique des points virgules** qui peut mener à des résultats très inattendu. Surtout lorsque vous débutez. Aussi prenez l'habitude de les mettre. Les outils comme Prettier les mettrons lors de la sauvegarde.

Mise à jour d'une variable

En JavaScript les types sont dynamiques, vous pouvez donc changer le type de valeur contenu par une variable sans problème :

```
var nomDeLaVariable;  
nomDeLaVariable = 1;  
nomDeLaVariable = [1, 2, 3];  
nomDeLaVariable = 'Paul';  
console.log(nomDeLaVariable); // 'Paul'
```

Copier

Recommandations sur les noms des variables

Les caractères acceptés pour les noms sont les chiffres, les lettres latines (A-Z) et les underscores (_).

Il ne faut pas utiliser de chiffre en début de nom.

Il est **fortement recommandé** d'utiliser le `lowerCamelCase` pour les variables.

A savoir : pas de majuscule en début de nom (sauf dans certains cas comme nous le verrons), puis une majuscule à chaque nouveau mot du nom.

Autre recommandation : **n'utilisez jamais des noms qui n'ont aucune signification (a, n etc). Soyez toujours explicite sur le contenu de la variable.**

Vous ne pouvez pas utiliser comme nom tous les mots clés réservés du JavaScript : par exemple `var`, `for`, `class`, `Document` etc.