



JavaScript - Variables et constantes

[Mise à jour le 19/6/2021]

- **Sources** et compléments sur MDN Web Docs
 - [Types et grammaire](#)
 - [Console JavaScript](#)
- **Lectures connexes**
 - Wikis WebPEM : ["Préparer un projet de site Web avec l'IDE VSCode"](#)

1. Variables

- **Sources** et compléments : [MDN web docs Mozilla](#)

Dans tous les langages de programmation, les variables permettent de **mémoriser des informations** dans le but de leur appliquer un traitement.



Une variable est identifiée par un **nom** et un **type**. Elle contient une **valeur**.

En JavaScript le nom d'une variable commence par une **lettre** ou par **\$**. Une variable est déclarée explicitement par le mot-clé **var** (ou **let** : voir le pour aller plus loin).

Il est possible, mais pas obligatoire d'initialiser une variable (lui attribuer une valeur) lors de sa déclaration.

1.1 Nommer une variable



Le nom d'une variable doit être représentatif de son contenu. Il ne doit pas contenir d'espaces ou de caractères accentués.



Exemple

`*.js`

```
var maMoyenneEnTsin ;
```



Dans la notation **camel case**, très utilisée en informatique, les mots constituant le nom d'une variable commencent par une majuscule, généralement à l'exception du premier.

1.2 Déclarer (créer) une variable



La déclaration des variables est **INDISPENSABLE**. Une variable est déclarée avec le mot-clé **var**. Utiliser une variable non déclarée entraîne une erreur à l'exécution.

Exemples

*.js

```
var i ; // variable non initialisée
var etat = true ; // type booléen
var $Somme = 400 ; // type entier
var maMoyenneEnTsin = 12.5 ; // type nombre réel
var monPrenom = "Lucas" ; // type chaîne de caractères
```



Pour déclarer **plusieurs** variables, les séparer par une virgule.

1.3 Les types de variables

Le JavaScript dispose actuellement de 7 types de données dont 6 sont dits primitifs:

- **Number** (les nombres),
- **String** (les chaînes de caractères),
- **Boolean** (les booléens),
- **Null** (rien, attribué à une variable pour qu'elle ne soit pas undefined),
- **undefined** (pas défini) et
- **Symbol** (disponible depuis ES6, est un type de données unique et immuable),
- plus le type **Object** (Objet, peut contenir plusieurs variables de types différents).



Le langage JavaScript est faiblement typé, c'est-à-dire qu'il autorise une variable à contenir n'importe quel type et à en changer en cours d'exécution.

- **Tester l'existence d'une variable avec typeof**



typeof est utilisé pour vérifier l'existence d'une variable ou en vérifier le type.

Exemple

*.js

```
var nombre = 20;
console.log(typeof nombre); // résultat dans la console : "number"

var texte = 'Mon texte';
console.log (typeof texte); // résultat dans la console : "string"

var booleen = false;
console.log (typeof booleen); // résultat dans la console : "boolean"

var test = null;
console.log (typeof test); // résultat dans la console : "object"

console.log (typeof nothing); // résultat dans la console : "undefined"
```

1.4 Les blocs d'instruction



En JavaScript, une instruction se termine par un **point-virgule** ou un **saut de ligne**. Un bloc d'instructions commence par **{** et se termine par **}**.

Exemple

.js

```
if (a >= 5) { // Exécuté si a>=5
  b -= 1;
  c += 2;
} else { // Exécuté si a<5
  b += 1;
  c -= 2;
}
```

1.5 La portée d'une variable



La portée d'une variable est l'ensemble des éléments du code source où la variable existe et est manipulable.

- **Variable locale** : une variable locale se déclare à l'intérieur d'une fonction et n'est accessible que dans la fonction.
- **Variable globale** : une variable globale se déclare en début de script et est accessible par toutes les fonctions du script. Dans une fonction, une variable globale peut être appelée en utilisant le mot-clé **this** (this.mavariable).

Exemple

*.html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Portée des variables</title>
</head>

<body>
  <script type="text/javascript">
    var a = 2; // variable globale

    function testvar() {
      var b = 4; // variable locale
      a = 1; // ou this.a
      console.log("a= " + a); // renvoie a=1
    }
    testvar();
    console.log("b= " + b) // renvoie Uncaught ReferenceError: b is
not defined
  </script>
</body>

</html>
```

1.6 Pour aller plus loin ...

Utilisation de **let** : [MDN web docs - Les variables](#)

2. Constante

- **Sources** et compléments : [MDN web docs Mozilla](#)



Une constante est comparable à une variable qui ne peut pas être modifiée une fois



créée. Elle est déclarée par le mot-clé **const** et doit être obligatoirement initialisée.

*.js

```
const PI = 3.1415927;
```

From:

<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<http://webge.fr/dokuwiki/doku.php?id=web:javascript:fondamentaux:variables&rev=1628671105>

Last update: **2021/08/11 10:38**

