



# LES VARIABLES

## Déclaration de variables

En JavaScript, les variables sont déclarées à l'aide du mot clé var, let ou const.

```
// Déclaration d'une  
variable avec var  
var x = 10;
```

```
// Déclaration d'une  
variable avec let  
let y = "Hello";
```

```
// Déclaration d'une  
variable avec const  
const z = true;
```

La déclaration avec var est la méthode traditionnelle de définir les variables en JavaScript.  
Une variable définie avec var a une portée qui est soit globale, soit locale à une fonction.

La déclaration avec let est une amélioration de var introduite avec ES6 (ECMAScript 2015).  
Une variable définie avec let a une portée de bloc, ce qui signifie qu'elle est limitée au bloc, à la déclaration, ou à l'expression dans lequel elle est déclarée

const est aussi introduit avec ES6 et permet de définir une constante, c'est-à-dire une variable qui ne peut pas être réaffectée. Comme let, const a aussi une portée de bloc.  
Si vous essayez de modifier la valeur d'une constante, JavaScript renverra une erreur.

La différence entre `let` et `const` est que `let` permet de réaffecter la variable, tandis que `const` ne le permet pas. Par exemple :

```
let x = 10;  
x = 20; // OK, x peut être réaffecté  
  
const y = 30;  
y = 40; // Erreur : y ne peut pas être réaffecté
```



## Types de données

JavaScript est un langage dynamiquement typé, ce qui signifie que les variables peuvent contenir n'importe quel type de données. Les types de données les plus courants en JavaScript sont :

- Les chaînes de caractères (string) : par exemple, "Hello, world!".
- Les nombres (number) : par exemple, 42 ou 3.14.
- Les booléens (boolean) : true ou false.
- Les tableaux (array) : une collection ordonnée de valeurs. Par exemple, `let arr = [1, 2, 3];`.
- Les objets (object) : une collection de propriétés et de valeurs. Par exemple, `let obj = { name: "John", age: 30 };`.
- Les valeurs nulles (null) et indéfinies (undefined).



Il est important de noter que les variables en JavaScript peuvent changer de type de données au cours du temps. Par exemple :

```
let x = "Hello"; // x est une chaîne de caractères  
x = 42; // x est maintenant un nombre
```

### Opérateurs d'affectation

En JavaScript, les variables peuvent être affectées avec des opérateurs d'affectation :

```
let x = 10;  
x += 5; // équivaut à x = x + 5  
x -= 3; // équivaut à x = x - 3  
x *= 2; // équivaut à x = x * 2  
x /= 4; // équivaut à x = x / 4
```



Il existe également des opérateurs d'incrémentation (++) et de décrémentation (--):

```
let x = 10;  
x++; // équivaut à x = x + 1  
x--; // équivaut à x = x - 1
```

### Concaténation de chaînes de caractères.

En JavaScript, les chaînes de caractères peuvent être concaténées à l'aide de l'opérateur '+' :

```
let firstName = "John";  
let lastName = "Doe";  
let fullName = firstName + " " + lastName; // équivaut à "John Doe"
```



## Utilisation de variables

Les variables peuvent être utilisées dans des expressions, des instructions de contrôle de flux (comme les if et les for), des fonctions et bien plus encore.

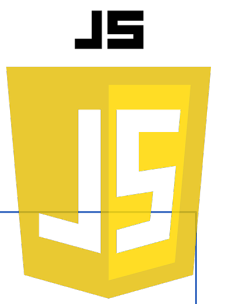
```
let x = 10;
let y = 20;
let z = x + y; // z vaut 30

if (z > 25) { console.log("z est supérieur à 25");}
else { console.log("z est inférieur ou égal à 25");}

for (let i = 0; i < 5; i++) {console.log(i);}

function sayHello(name) {
  console.log("Hello, " + name + "!");
}

sayHello("John"); // affiche "Hello, John!"
```



## Portée des variables:

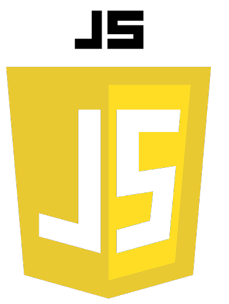
La portée d'une variable fait référence à la partie du code où une variable peut être référencée ou accessée. JavaScript a deux types de portée : globale et locale.

- **Portée globale** : Une variable déclarée en dehors d'une fonction a une portée globale et peut être accessée de n'importe où dans le code.
- **Portée locale** : Une variable déclarée à l'intérieur d'une fonction a une portée locale et ne peut être accessée qu'à l'intérieur de cette fonction.

## Nommer les variables :

Il y a quelques règles à suivre lors de la dénomination des variables en JavaScript :

- Les noms de variables sont sensibles à la casse (y et Y sont deux variables différentes)
- Les noms de variables doivent commencer par une lettre, un dollar (\$) ou un underscore (\_)
- Les noms de variables ne peuvent pas commencer par un chiffre
- Les noms de variables ne peuvent pas contenir d'espaces ou de caractères spéciaux, à l'exception du dollar (\$) et de l'underscore (\_).



Ci-joint un extrait de code qui démontre l'utilisation des variables en JavaScript. Vous pouvez le consulter et l'exécuter en suivant ce lien :

<https://app.coderpad.io/sandbox?snippet=79973d1a>

Cet extrait de code illustre la déclaration, l'attribution et l'utilisation de variables en JavaScript. Il montre également comment utiliser différents types de données, la portée des variables, et comment nommer correctement les variables.