

### Variable

automatique	nomVar = "data"
let	let nomVar = "data"
var	var nomVar = "data"
const	const nomVar = "data"

Toujours utiliser const si la valeur ou le type (Tableaux et Objets) ne doit pas être modifiée

N'utilisez que let si vous ne pouvez pas l'utiliser const

À utiliser uniquement var si vous DEVEZ prendre en charge les anciens navigateurs

### Commentaire

commentaire sur une ligne	// commentaire
Commentaires plusieurs lignes	/* commentaires */

### Opérateurs

+	Addition
-	Soustraction
*	Multiplication
/	Division
++	Incrément
--	Décrément
+=	Additionne puis affecte le résultat à la variable
-=	Soustrait puis affecte le résultat à la variable
%	Modulo
**	Puissance

### Fonction nombre

number.to-String()	renvoie un nombre sous forme de chaîne
Number.parseInt()	utilisée pour convertir des variables ("10") JavaScript en nombres

### Fonction nombre (cont)

Number.isInteger()	Renvoie vrai si l'argument est un entier
Number.isSafeInteger()	Renvoie vrai si l'argument est un entier sûr
Number.parseFloat()	Convertit une chaîne en nombre Float()
Number.parseInt()	Convertit une chaîne en nombre entier

### Switch

```
let x = "0";
switch (x) {
  case 0:
    text = " Off ";
    break;
  case 1:
    text = " On";
    break;
  default:
    text = "No value found";
}
```

### Boucle for

```
for (let i = 0; i < 5; i++) {
  text += i;
}
```

L'expression 1 définit une variable avant le début de la boucle (soit i = 0).

L'expression 2 définit la condition d'exécution de la boucle (i doit être inférieur à 5).

L'expression 3 augmente une valeur (i++) à chaque fois que le bloc de code de la boucle est exécuté.

.length

### foreach

```
const numbers = [45, 4, 9, 16, 25];
let txt = " ";
numbers.forEach(myFunction);
function myFunction(value, index, array) {
  txt += value;
}
```

La valeur de l'article, L'index des articles  
Le tableau lui-même

### Classe

créer une class

```
class Car {
  constructor(name, year) {
    this.name = name;
    this.Year = year;
  }
}
```

utiliser une class

```
const myCar1 = new Car("Ford", 2014);
```

créer des méthodes

```
class ClassName {
  constructor() { ... }
  method_1() { ... }
  method_2() { ... }
  method_3() { ... }
}
```

### Classe

créer une class

```
class Car {
  constructor(name, year) {
    this.name = name;
    this.Year = year;
  }
}
```

### Classe (cont)

```
> utiliser une class
const myCar1 = new Car("Ford", 2014);
créer des méthodes
class ClassName {
  constructor() { ... }
  method_1() { ... }
  method_2() { ... }
  method_3() { ... }
}
```

### Fonction

```
function nomFonction(a, b) {
  return a * b;
}
// utilisation de la fonction
let x = myFonction(4, 3);
```

### Chaîne de caractère

```
let Name1 = "john"; // Double quotes
let Name2 = 'doe'; // Single quotes
```

Le caractère d'échappement barre oblique inverse ( \ ) transforme les caractères spéciaux en caractères de chaîne

### fonction chaîne caractère

text.length	renvoie la longueur d'une chaîne
text.charAt(0)	renvoie le caractère qui est à cette index
text.charCodeAt(0)	renvoie le code utf
text.slice(7, 13)	Découpez une partie d'une chaîne de la position 7 à la position 13
text.toUpperCase()	chaîne est convertie en majuscule

### fonction chaîne caractère (cont)

text.Trim()	supprime les espaces des deux côtés d'une chaîne
text.trimStart()	supprime les espaces uniquement à partir du début
text.trimEnd()	supprime les espaces uniquement à la fin
text.padStart(4,"0")	méthode remplit une chaîne depuis le début
text.padEnd(4,"0")	méthode remplit une chaîne à partir de la fin
text.repeat(2)	méthode renvoie une chaîne avec un certain nombre de copies d'une chaîne
text.replace("Microsoft", "W3Schools")	méthode remplace une valeur spécifiée par une autre valeur dans une chaîne
text.match("ain")	renvoie un tableau contenant les résultats de la correspondance
text.includes("world")	includes() méthode renvoie true si une chaîne contient une valeur spécifiée

### for in

```
const person = {
  fname: "John", lname: "Doe", age: 25};
let text = "";
for (let x in person) {
  text += person[x];
}
const numbers = [45, 4, 9, 16, 25];
let txt = "";
for (let x in numbers) {
  txt += numbers[x];
}
```

La boucle for in parcourt un objet personne  
Chaque itération renvoie une clé (x)  
La clé permet d'accéder à la valeur de la clé  
La valeur de la clé est personne[x]

### Condition

```
if (condition) { instruction }
if (condition) { instruction } else { instruction }
if (condition1) { instruction } else if (condition2) { instruction } else { instruction }
```

### For of

```
const cars = ["BMW", "Volvo", "Mini"];
let text = "";
for (let x of cars) {
  text += x;
}
```

for of parcourt les valeurs d'un objet itérable.

Il vous permet de parcourir des structures de données itérables telles que des tableaux, des chaînes, des cartes, des listes de nœuds, etc



### Fonction math

Math.abs(a)	Retourne la valeur absolue de a
Math.round(a)	Retourne l'entier arrondi le plus proche de a
Math.ceil(a)	Retourne l'entier immédiatement supérieur (ou égal) à a
Math.floor(a)	Retourne l'entier immédiatement inférieur (ou égal) à a
Math.sqrt(a)	Retourne la racine carrée de a
Math.log(a)	Retourne le logarithme de a
Math.ln(a)	Retourne le logarithme népérien de a
Math.exp(a)	Retourne l'exponentielle de a
Math.pow(a,b)	Retourne a à la puissance b
Math.min(a,b)	Retourne le plus petit des paramètres a ou b
Math.max(a,b)	Retourne le plus grand des paramètres a ou b

### Objet

```
const person = {
  prenom : " John",
  nom : " Doe ",
  id : 9985,
  fullName : function() {
    return this.nom + " " +
    this.prenom;
  }
};
// 2 utilisations possible
person.nom;
person ["pre nom"];
```

Dans une méthode objet, this fait référence à l'objet .

### Type

String	Chaîne de caractères	let test = "Hello world !";
Number	Nombre	let test = 42;
Boolean ou bool	Vrai/Faux	let test = true;
Null	Nul ou vide	let test = null;
Undefined	non définie	

### Array

Créer un tableau	const cars = ["Saab", "Volvo", "BMW"];
Créer un tableau	const cars = []; cars[0]= "Saab"; cars[1]= "Volvo"; cars[2]= "BMW";
Créer un tableau	const cars = new Array("Saab", "Volvo", "BMW");

### Fonction Array

let length = fruits. Length;	longueur d'un tableau (le nombre d'éléments du tableau)
tab.push("data");	ajouter un nouvel élément à un tableau
Array.isArray(fruits)	renvoie true si un tableau
tab.toString()	convertit un tableau en chaîne
tab.join(" * ");	comme toString(), mais spécifier le séparateur
tab.pop()	supprime le dernier élément
tab.shift();	supprime le premier élément du tableau

### Fonction Array (cont)

tab.unshift("data");	ajoute un nouvel élément à un tableau (au début)
tab1.concat(tab2);	crée un nouveau tableau en fusionnant
tab.indexOf("data");	recherche dans un tableau une valeur d'élément renvoie sa position
tab.sort()	trie par ordre alphabétique
tab.reverse()	inverse les éléments
tab.toSorted()	comme sort mais crée un nouveau tableau, en gardant le tableau d'origine inchangé
tab.toReversed()	comme reverse mais crée un nouveau tableau

### Comparaison

==	vérifie l'égalité de deux valeurs de manière laxiste
===	vérifie l'égalité de deux valeurs de manière stricte.
<=	valeur est inférieure ou égal
>=	valeur est supérieure ou égal
!=	tester la différence en valeurs
!==	tester la différence en valeurs ou en types
<	valeur est strictement inférieure
>	valeur est strictement supérieure

### while

```
while (condition) {instruction}
```

```
do { instruction } while (condition);
```

while boucle parcourt un bloc de code tant qu'une condition spécifiée est vraie  
do while boucle est une variante de la boucle while. Cette boucle exécutera le bloc de code une fois, avant de vérifier si la condition est vraie, puis elle répétera la boucle tant que la condition est vraie.

### Fonction fléchée

```
hello = () => {  
  return "Hello World!";  
}
```

Les fonctions fléchées renvoient la valeur par défaut

```
hello = () => "Hello World!";
```

Fonction flèche avec paramètres

si vous n'avez qu'un seul paramètre, vous pouvez également ignorer les parenthèses

```
hello = (val) => "Hello " + val;
```

Fonction flèche sans parenthèses

```
hello = val => "Hello " + val;
```

Si la fonction n'a qu'une seule instruction et que l'instruction renvoie une valeur, vous pouvez supprimer les crochets et le return-mot-clé



By **mr.freelancer**

[cheatography.com/mr-freelancer/](https://cheatography.com/mr-freelancer/)

Not published yet.

Last updated 15th March, 2024.

Page 4 of 4.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>