

JavaScript

Petit guide du codeur

AMATHIEU Valentin

NOTES IMPORTANTES !

Ce document ne se veut pas être une référence exhaustive de la syntaxe et des fonctionnalités du Javascript.

Il s'agit d'une reprise synthétique du cours. Il est néanmoins agrémenté de notions supplémentaires très utilisées pour développer rapidement une application dans ce langage.

Vous êtes encouragés à approfondir vos connaissances si besoin est, à l'aide de ressources telles que :

- <https://developer.mozilla.org/fr/docs/Web/JavaScript>
- <https://www.w3schools.com/js/>

Environnements d'exécution

- **Côté client** : Navigateur web (via la page web ou la console javascript)

Exemple(s) : Mozilla Firefox, Google Chrome, Opéra, Safari, IE/MS Edge

- **Côté serveur** : Terminal/Console

Exemple(s) : NodeJS

Javascript & page web

```
<!DOCTYPE html>
<html>
  <head>
    <title>Guide du codeur</title>
    <meta charset="utf-8">

    <!-- Référence à un script situé sur le web -->
    <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>

    <!-- Référence à un script situé en interne -->
    <script src="js/utilisateur/utilisateur.js"></script>

    <!-- Code directement mis dans la page -->
    <script type="text/javascript">
      let age = 18;
      console.log("J'ai " + age + " ans");
    </script>
  </head>

  <body>
  </body>
</html>
```

Les variables

- Ancienne notation :

```
var age = 18;
```
- Nouvelle notation :

```
let age = 18;
```
- Constante :

```
const age = 18;
```

Les tableaux

```
let technique = ["tremette", "charge", "ultralaser"];  
console.log("Leviator ! Attaque " + technique[0].toUpperCase() + " !!!");
```

Leviator ! Attaque TREMPETTE !!!

Les conditions

- If ... else if ... else

```
if (age < 3) {  
    console.log("Tu es un bébé");  
} else if (age < 12) {  
    console.log("Tu es un enfant");  
} else if (age < 18) {  
    console.log("Tu es un adolescent");  
} else {  
    console.log("Bonjour Monsieur");  
}
```

- Switch ... case ... default

```
switch (langue) {  
    case "fr":  
        console.log("Bonjour !");  
        break;  
    case "en":  
        console.log("Hello !");  
        break;  
    case "es":  
        console.log("Hola !");  
        break;  
    default:  
        console.log("...");  
        break;  
}
```

Les boucles

Tous ces exemples produiront le même résultat !

- While

```
let i = 0;
while (i < 10) {
  console.log(i);
  i++;
}
```

- Do ... while

```
let i = 0;
do {
  console.log(i);
} while (i < 10);
```

- For

```
for (let i = 0 ; i < 10 ; i++) {
  console.log(i);
}
```

- For ... in

```
let num = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
for (let i in num) {
  console.log(num[i]);
}
```

- For ... of

```
let num = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
for (let i of num) {
  console.log(i);
}
```


Les fonctions

```
function attaque (nomAttaque) {  
    console.log("Leviator ! Attaque " + nomAttaque.toUpperCase() + " !!!");  
}  
  
attaque("tremette");
```

Leviator ! Attaque TREMPETTE !!!

Les objets

{ ... } : Notation raccourcie de la création d'un objet en JS
L'objet est implicitement instancié à partir de la classe native 'Object'

```
let pokemon = {  
  nom: "Leviator",  
  technique: ["tremquette", "charge", "ultralaser"],  
};  
  
console.log(pokemon.nom + " ! Attaque " + pokemon.technique[0].toUpperCase() + " !!!");
```

Leviator ! Attaque TREMPETTE !!!

Les classes

Permet de créer des objets à partir de schémas prédéfinis et personnalisés

```
class pokemon {  
  
  constructor(nom) {  
    this.nom = nom;  
    this.skill = [];  
  }  
  
  learnSkill(skill) {  
    this.skill.push(skill);  
  }  
  
  listAllSkill() {  
    console.log(this.nom + " possède toutes ces techniques :\n- " + this.skill.join("\n- "));  
  }  
}  
  
let pikachu = new pokemon("pikachu");  
pikachu.learnSkill("éclair");  
pikachu.learnSkill("fatal foudre");  
pikachu.listAllSkill();
```

pikachu possède toutes ces techniques :

- éclair
- fatal foudre

Objets natifs : Number

```
/*  
    La valeur de num étant un nombre, ce dernier est implicitement  
    transformé en objet de la classe 'Number'  
    L'objet num bénéficie donc des fonctions natives de manipulation  
    des nombres  
*/  
var num = 10.7689;  
  
// 'toFixed(x)' renvoie le nombre avec x décimales après la virgule  
// un arrondi est effectué sur la dernière décimale  
console.log(num.toFixed(2)); // 10.77  
  
// 'toPrecision(x)' renvoie les x premiers chiffres d'un nombre  
// un arrondi est effectué sur le dernier chiffre  
console.log(num.toPrecision(3)); // 10.8
```

Objets natifs : String

```
/*
  La valeur de chaine étant une chaine de caractère, cette dernière est implicitement
  transformé en objet de la classe 'String'
  L'objet chaine bénéficie donc des fonctions natives de manipulation
  des chaines de caractère
*/
var chaine = "Yes we can !";

// 'charAt(x)' renvoie le caractère situé à l'indice x dans la chaîne
console.log(chaine.charAt(2)); // 's'

// 'charAt(x)' renvoie le caractère situé à l'indice x dans la chaîne
console.log("B.O. : ".concat(chaine)); // 'B.O. : Yes we can !'

// 'indexOf(x)' renvoie l'indice de la première occurrence de x dans la chaîne
console.log(chaine.indexOf("e")); // 'B.O. : Yes we can !'

// 'match(x)' indique si l'expression régulière 'x' a une/des correspondance(s) dans la chaîne (voir regexp)
console.log(chaine.match(/we/)); // ["we"]

// 'replace(x, y)' remplace la première occurrence de x par y
// si une expression régulière est utilisée (avec le modificateur 'g') dans x,
// alors toute occurrence de x peut être remplacée par y
console.log(chaine.replace("we", "you")); // 'Yes you can'
console.log(chaine.replace(/\s/g, "+")); // 'Yes+we+can+!'
```

Objets natifs : String

```
/*  
  La valeur de chaine étant une chaine de caractère, cette dernière est implicitement  
  transformé en objet de la classe 'String'  
  L'objet chaine bénéficie donc des fonctions natives de manipulation  
  des chaines de caractère  
*/  
var chaine = "Yes we can !";  
  
// 'split(x)' découpe la chaîne en un tableau de sous-chaîne dont le délimiteur est x  
// Par défaut le délimiteur est ""  
console.log(chaine.split()); // ["Y", "e", "s", " ", "w", "e", " ", "c", "a", "n", " ", "!"]  
console.log(chaine.split(" ")); // ["Yes", "we", "can", "!"]  
  
// 'substr(x, y)' extraie de la chaîne, une sous-chaîne de y caractères, en commençant à l'indice x  
console.log(chaine.substr(7, 3)); // "can"
```

Objets natifs : Math

```
Math.abs(-2); // 2  
Math.ceil(1.6) // 2  
Math.floor(2.8) // 2  
Math.max(3, 5, 89, 34, 2); // 89  
Math.min(3, 5, 89, 34, 2); // 2  
Math.random() // valeur aléatoire entre 0 et 1, ex : 0.762478
```

Objets natifs : Array

```
let tab = [2, 3, 7, 1, 4];  
// Concaténation de 2 tableaux  
console.log(tab.concat([5, 8])); // [2, 3, 7, 1, 4, 5, 8]  
// Renvoie l'index d'un élément du tableau  
console.log(tab.indexOf(7)); // 2  
// Ajoute un élément à la fin du tableau  
tab.push(8);  
// Retourne le dernier élément du tableau et le retire de celui-ci !  
console.log(tab.pop()); // 8  
// Ajout un élément au début du tableau  
tab.unshift(9);  
// Retourne le premier élément du tableau et le retire de celui-ci !  
console.log(tab.shift()); // 9
```


Objets natifs : Array

```
let tab = [3, 7, 1];

// Renverse l'ordre de tous les éléments du tableau
console.log(tab.reverse()); // [1, 7, 3]
// Trie les éléments du tableau dans l'ordre ascendant
console.log(tab.sort()); // [1, 3, 7]

// Parcourt les éléments du tableau
tab.forEach(function (element, indice) {
    console.log(indice + " -> " + element);
});
/*
0 -> 1
1 -> 3
2 -> 7
*/
```