



×



JavaScript

&

jQuery

Date de première version
4 décembre 1995

JavaScript



JavaScript est un langage de programmation de **scripts** principalement employé dans les **pages web** interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js.

JavaScript

C'est un langage **orienté objet à prototype**, c'est-à-dire que les **bases** du langage et ses principales interfaces sont fournies par des **objets** qui ne sont pas des instances de classes, mais qui sont chacun équipés de **constructeurs** permettant de créer leurs propriétés.

L'utilisation de JavaScript

JavaScript arrive toujours dans le **trio de tête des langages de programmation** les plus utilisés, car il propose une **multitude** de fonctionnalités au choix et des **frameworks** utilisé dans différents domaines.

Il est difficile de s'en passer pour le développement web tant il est complet.

JavaScript auprès des entreprises

La **grande majorité** des développeurs **savent** coder dans ce langage. Il ne suffit donc plus de mentionner celui-ci dans son CV pour faire la **différence avec d'autres développeurs sur le marché.**

En revanche il est **essentiel de le mentionner** dans son C.V. mais surtout de **maîtriser** les librairies et/ou frameworks les plus utilisés en JavaScript (React.js, Vue.js, AngularJS, Node.js, Express.js ...)

Exemple de librairie Front pour JavaScript



Afin de réaliser des **animations web** facilement, plusieurs outils sont disponibles. L'un des plus connus est **GreenSock**. Le site officiel est [ici](#), on peut aussi voir un exemple de l'utilisation de la librairie [ici](#).

Exemple de librairie de mapping en JS



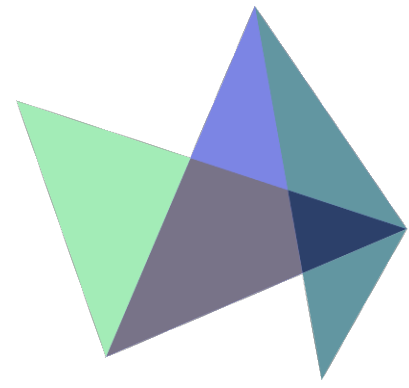
L'intégration de cartes interactives et compatibles à la fois sur mobile et desktop devient plus facile grâce à la librairie Open Source **leaflet.js**.

Elle intègre des cartes OpenStreetMap (Open Source) et la manipulation de ces dernières se fait facilement en raison d'une facilité d'utilisation de la librairie.

Exemple de librairie de jeux pour JavaScript



Pour réaliser des **jeux web** simplement, plusieurs outils sont disponibles. L'un des plus utilisé est **Phaser**. Le site officiel est [ici](#), on peut aussi voir un exemple de jeu web créé à l'aide de la librairie [ici](#).



Exemple de librairie de stats pour JavaScript



Pour créer des graphiques bien désignés de manière très simple, il existe des libraires JavaScript comme **Highcharts.js** ou **chart.js**.

Date de première version
26 août 2006

jQuery



jQuery est une **bibliothèque JavaScript** libre et multiplateforme créée pour **faciliter l'écriture** de scripts côté client dans le code HTML des pages web

L'utilisation de jQuery


















LIBRAIRIES JS

Nom	Tout le web	Top 10k	Top 100k	Top Million
jQuery	34,1%	12,4%	17,1%	20,1%
Google Libraries	6,8%	6,9%	7,8%	5,2%
html5shiv	6,2%	5,9%	6,4%	5,6%
jQuery UI	4,9%	5,1%	6,0%	6,7%
Facebook for Websites	4,6%	7,3%	7,3%	7,0%
Modernizr	3,6%	5,5%	5,4%	3,9%
jQuery Form	3,5%	1,6%	2,0%	2,7%
Facebook SDK	3,3%	6,1%	6,0%	5,7%
jQuery Easing	3,2%	1,5%	2,0%	2,5%
Fancybox	3,0%	2,0%	2,3%	3,0%
jQuery Cycle	2,3%	1,6%	2,1%	2,4%
Lightbox	2,1%	1,3%	1,7%	2,5%
FlexSlider	1,8%	1,2%	1,5%	1,6%
SuperFish	1,7%	0,8%	1,1%	1,6%

Source : [Builtwith](#)

Date : 2015

Top In JavaScript Usage Distribution in the Top 1 Million Sites

Technology	Websites	%
 jQuery	783,208	78.32
 Facebook for Websites	328,778	32.88
 Facebook SDK	302,554	30.26
 Google Hosted Libraries	258,991	25.9
 html5shiv	258,054	25.81
 Modernizr	233,214	23.32
 jQuery UI	229,064	22.91
 Bootstrap.js	152,071	15.21
 Respond	128,948	12.89
 Twitter Platform	114,830	11.48
 jQuery Cookie	107,058	10.71
 jQuery Masonry	106,920	10.69
 imagesLoaded	103,547	10.35
 Fancybox	103,471	10.35
 Google API	100,416	10.04
 yepnope	92,818	9.28
 Magnific Popup	89,423	8.94
 Slick JS	84,040	8.4
 FitVids.JS	81,135	8.11

Date : 2018

jQuery



Il faut savoir que l'utilisation faite des **librairies** et des **frameworks** JavaScript **évolue énormément**. Actuellement, jQuery est **très utilisé**.

Pourquoi jQuery

Depuis sa création en **2006**, jQuery a connu un large succès auprès des développeurs Web et son apprentissage est aujourd'hui un des **fondamentaux** de la formation aux **technologies du Web**.

jQuery est à l'heure actuelle la librairie front-end **la plus utilisée au monde** (plus de la moitié des sites Internet en ligne intègrent jQuery).

Frameworks JS

Les **frameworks** JavaScript les plus utilisés sont **React.js**, **AngularJS** et **Vue.js**. Ils succèdent aux frameworks plus anciens tels que ember.js, meteor.js, backbone.js.

Les langages de programmation évoluent très rapidement, et il est important de savoir **s'auto-former** et **s'adapter** aux nouveautés. C'est pourquoi avoir de bonnes bases de programmation en JavaScript est très important.

Frameworks JS

D'après le site **alsacreations.com** :

- **React.js** : passe de 48% à 64% (raisons : style de programmation élégant, écosystème de paquets riche).
- **Vue.js** : passe de 6% à 28% (raisons : courbe d'apprentissage plus aisée, style élégant, bonne documentation, léger).
- **AngularJS** : stagne à environ 20%.

JavaScript

Revenons dans le concret : **JavaScript_D01**
Concentrons-nous sur les bases de ce nouveau langage

Votre meilleur ami

Le **Mozilla Developer Network** : le **MDN**. Il contient toute la documentation donc vous avez besoin. Autrement il y a toujours **stackoverflow** pour vous aider à **résoudre vos problèmes** spécifiques en **JavaScript**

Les commentaires

// Les commentaires sont comme en C.

// Les commentaires **mono-ligne commencent** par **2 slashes**

/* et les commentaires sur **plusieurs lignes commencent** avec **slash-étoile**
et **finissent** avec **étoile-slash */**

Les fins d'instructions

Les expressions (ligne de code) peuvent se finir par un « ; » comme en C, mais ceci **n'est pas obligatoire**.

Néanmoins, si vous voulez être « **à la norme** » je vous invite à les mettre.

Les int(s)

JavaScript n'a qu'un **seul** type d'**int**, il n'y pas pas de long double, de float, d'unsigned int ... etc.

Ainsi :

- $7 / 2 = 3.5$
- $(15 + 5) * 2 = 42$
- $42 / 0 = \text{Infinity}$
- $0 / 0 = \text{NaN}$ (**Not a Number**)

Les booléens

Comme dans le langage **C** :

- true
- false
- !false = true
- !true = false

Les comparaisons

- `42 == 42` (`true`)
- `42 == 21` (`false`)
- `42 != 42` (`false`)
- `42 != 21` (`true`)

- `42 > 21` (`true`)
- `42 < 21` (`false`)
- `42 <= 42` (`true`)
- `42 >= 42` (`true`)
- `42 >= 21` (`true`)

La concaténation

```
var str = 'Hello' + ' ' + 'World';
```

```
// str = "Hello World";
```

Déclaration de variable

Mot-clés : **var** / **let** / **const**

Possibilité de déclarer **plusieurs variables** sur une seule et même ligne :

- var s1 = "toto", s2 = "tata";

var : scope global au contexte

let : scope local au contexte
(disponible qu'en ES6)

const : scope global à valeur
immuable

Créer un Array / Objet simplement

```
var tab = [1,2]; // Tableau de 2 valeurs  
tab[0] = 1;
```

```
var obj = {"key" : 4} // Objet avec 1 valeur  
obj["key"] = 4;
```


Exemple de parcours d'objet

```
var person = {fname:"Sebastien", lname:"Roth", age:26};

for (var x in person) {
    description += person[x] + " ";
}
console.log(description); // "Sebastien Roth 26"
```

let ou var

```
function varTest() {  
  var x = 1;  
  if (true) {  
    var x = 2; // même variable  
    console.log(x); // 2  
  }  
  console.log(x); // 2  
}  
  
function letTest() {  
  let x = 1;  
  if (true) {  
    let x = 2; // variable différente  
    console.log(x); // 2  
  }  
  console.log(x); // 1  
}
```

- **Var** - Créé une variable utilisable accessible **n'importe** où dans la **fonction varTest()**. Elle n'existe pas dans letTest()
- **Let** - Créé une variable **dans le scope** où elle est **définie**, dans l'exemple, le x avec la valeur 2 est **uniquement** accessible dans le if

Fun facts JavaScript

```
// fonction à écrire sur votre IDE
function add(a, b) {
    var res = a + b;
    console.log(res);
}

add(0.1, 0.2);
```

Pour connaître plus de spécificités JavaScript, cliquez [ici](#) !