



## Chapitre 1 : Généralité sur Javascript

**Objectifs :** Les objectifs sont avant tout de vous présenter le fonctionnement du JavaScript, vous présenter les possibilités offertes par ce langage et comment il s'intègre dans un site web.

**Prérequis :** Avant d'entamer cette séquence, vous n'avez besoin d'aucune connaissance préalable en JavaScript, mais vous devriez être familier avec HTML et CSS

### 1. Historique

JavaScript est un langage très controversé. Au début de l'Internet, les pages étaient constituées uniquement de textes et de liens hypertextes, ce qui restreignait l'usage à des scientifiques et des universitaires. De toutes manières, les contraintes techniques de l'époque, notamment au niveau des débits de transfert, ne pouvaient pas proposer autre chose de mieux. C'est au milieu des années 1990 que le besoin de disposer de sites Internet plus conviviaux et proposant plus de services est apparu. Brendan Eich, alors ingénieur informaticien chez Netscape est chargé du développement d'un nouveau navigateur web. Il en profite pour développer un langage de Script, à l'origine nommé LiveScript, et qui devait être un complément à Java (ces deux langages sont souvent confondus du fait de leur appellation quasi-identique, bien qu'ils n'aient que peu de choses en commun). L'objet de ce langage de script est de rendre les pages Internet plus attractives, plus conviviales pour le visiteur, en permettant davantage de choses sans pour autant faire appel à la programmation côté serveur. Pour ce faire, le navigateur doit pouvoir interpréter le code JavaScript. Netscape décide d'implémenter nativement LiveScript dans la version 2.0 de son navigateur (alors baptisé Netscape Navigator) dès 1995. Débute alors, une période de grande popularité des langages de script et Microsoft ne pouvait que se résoudre à sortir sa propre version. Ce fut Jscript sorti en 1996 et intégré à son

navigateur Internet Explorer dont la dernière version est aujourd'hui Jscript.Net. Les versions de JavaScript se sont alors succédées, apportant pour chacune d'entre elles son lot d'améliorations. Tout le monde a pu constater que l'Internet a rapidement été envahi de pages comportant de petits scripts permettant, par exemple, d'afficher l'heure, la date, le nom du visiteur, ou effectuant la validation du contenu de champs de formulaire. Cependant, même si JavaScript suit les instructions données par l'ECMA (European Computer Manufacturers Association), organisme international chargé de la standardisation des systèmes d'information et de communication, les éditeurs de logiciels (Microsoft d'un côté avec Internet Explorer et Sun de l'autre avec Firefox), ont élaboré, depuis le début, des navigateurs qui implémentent différemment JavaScript. De ce fait, certains scripts peuvent très bien s'exécuter normalement sur un navigateur et paradoxalement, générer une erreur sur un autre. C'est en partie à cause de cela qu'à la fin des années 1990, d'autres langages tels ASP ou PHP deviendront plus populaires. Mais c'est surtout l'utilisation à outrance de pop-up (fenêtre surgissant) qui est à l'origine de la baisse d'intérêt pour l'emploi de JavaScript. Leur prolifération a énormément nui à JavaScript et l'exaspération des utilisateurs a fini par en cacher les avantages aux yeux des développeurs ; certains allant même jusqu'à le considérer comme un sous langage. Heureusement, l'arrivée des bloqueurs de pop-up intégrés aux navigateurs a permis à JavaScript de redorer son blason.

## 2. Limites et avantages de JavaScript

JavaScript est un langage universel qui se retrouve dans de nombreuses pages HTML, en complément de ce code. Grâce à JavaScript, les pages HTML sont plus riches et disposent de nombreuses fonctionnalités supplémentaires.

Savoir rédiger des scripts en JavaScript, c'est permettre aux visiteurs de vos pages HTML d'accéder à d'autres fonctionnalités, à d'autres services et d'améliorer ainsi la professionnalisation d'un site. Ainsi, il y a encore quelques temps, lorsqu'un utilisateur choisissait un identifiant pour la première fois, il fallait cliquer sur un bouton et attendre une réponse de la part du serveur. Et, celui-ci, parfois, invitait à recommencer le processus de création, le pseudo étant déjà pris. Alors qu'aujourd'hui, avec l'emploi de la technologie AJAX, le contrôle s'effectue en arrière-plan pendant que le visiteur complète sa fiche. Il est indéniable que JavaScript contribue fortement à la convivialité d'un site Internet et améliore, par conséquent, la fidélisation des visiteurs.

Étant donné cette large diffusion, savoir rédiger des scripts JavaScript est devenu, aujourd'hui, une connaissance de base de tout développeur web.

Pour autant, l'usage de JavaScript n'est pas réservé au web, en effet plusieurs logiciels du marché tels Adobe Photoshop ou Adobe Illustrator utilisent des variantes très proches de JavaScript pour automatiser certaines tâches. En ce qui concerne la difficulté du langage, certains pourraient être réticents à l'analyse des pages HTML contenant du code JavaScript, mais finalement avec un peu de temps et de recherche, JavaScript est un langage dont la maîtrise est assez facile. D'autant plus si vous êtes déjà familier avec d'autres langages tels le Visual Basic ou le langage C, par exemple, même si une adaptation à quelques particularités est tout de même nécessaire.

À l'inverse, JavaScript ne peut pas tout faire. Comme c'est un langage de script qui s'exécute côté client, il lui est impossible de s'interfacer avec une base de données de type Mysql ou SQL, par exemple. Pour remplir cette mission, il faut impérativement passer par la programmation côté serveur avec des langages tels ASP ou PHP. Autre point important, JavaScript n'est pas en mesure d'écrire ou de lire sur le disque dur du poste client (hormis les cookies qui ne sont que de petits fichiers texte et dont nous traiterons les différents aspects au chapitre Améliorer l'interactivité avec JavaScript et CSS). Cette dernière limitation n'en constitue pas pour autant un défaut, car ainsi JavaScript ne propage pas d'infections virales fortement dangereuses.

Une autre particularité de JavaScript réside dans le fait qu'il ne nécessite pas d'éditeur particulier ni de compilateur. Il est très facile de rédiger des scripts directement dans le code de la page HTML en passant par un simple éditeur de texte de type Wordpad ou un éditeur de code HTML. Il existe, cependant, des outils de conception dont l'apport n'est pas négligeable.

### 3. Outils de conception

Pour apprendre et exploiter le Javascript, il vous faut :

- Un navigateur qui reconnaît le Javascript.
- Une solide connaissance du Html
- Un simple éditeur de texte

### 3.1. Un navigateur qui reconnaît le Javascript.

Un **navigateur Web** (ou **browser** en anglais) est un logiciel conçu pour afficher des pages Web. C'est le type de logiciel qu'on utilise pour surfer sur le Web. Tous les ordinateurs actuels, qu'ils soient fixes, portables ou mobiles, intègrent au moins un navigateur Web. Les plus connus sont **Firefox**, **Chrome** et **Internet Explorer**, mais il en existe bien d'autres.

### 3.2. Une solide connaissance du Html

Comme le code du Javascript vient s'ajouter au "code" du langage Html, une connaissance des balises ou tags Html est souhaitable. Je vous recommande un tutorial d'**openclassroom** pour apprendre le langage Html : <https://openclassrooms.com/courses/apprendre-asp-net-mvc/introduction-au-html> .

### 3.3. Un simple éditeur de texte

Les outils permettant d'insérer du code JavaScript sont nombreux. Cela va du simple logiciel éditeur de texte, comme par exemple WordPad de Windows, à l'outil spécifique Aptana Studio, en passant par les éditeurs de code HTML tels Dreamweaver ou Visual Studio Code, avec lesquels il est possible d'insérer des blocs JavaScript. L'usage de ces logiciels permet de disposer d'un certain nombre d'outils facilitant l'écriture du code. Il est, par exemple, fort simple de :

- Vérifier une syntaxe par la coloration automatique du code ;
- Disposer de l'autocomplétion (proposition des méthodes ou propriétés disponibles de l'objet) ;
- Connaître la valeur d'une variable lors de l'exécution d'un script.

Dans ce cours nous allons travailler avec le logiciel **notepad++** téléchargeable sur ce lien :

<https://notepad-plus-plus.org/fr/>

### 3.4. Paramétrages et environnement optimal de test

Pour débiter en JavaScript, un minimum de connaissances en HTML est nécessaire et notamment la notion de balise permettant de se situer dans la page. Pour mémoire, nous rappellerons simplement qu'une page HTML se divise en deux grandes parties :

- La partie **head** (**tête en français**) dans laquelle se situent les informations correspondant à la description du contenu ;

- La partie **body (corps en français)** où figure le code permettant la construction des objets dans la page (champs de formulaire, zone de texte, image, etc.).

Un script JavaScript peut se trouver au choix dans l'une ou l'autre de ces deux parties. Cependant, par convention, les scripts se retrouvent plus fréquemment dans la partie **head** de la page. Leur exécution peut alors être immédiate (au chargement de la page) ou différée (clic sur un bouton, par exemple). Il faudra, dans ce cas, utiliser la programmation événementielle et les fonctions pour que le code s'exécute. Ces points sont traités au chapitre Fonctions et événements du présent ouvrage. La position des scripts dans la partie **head** ne signifie pas pour autant que les scripts seront indexés par les moteurs de recherche. En effet, jusqu'à présent, les moteurs tels Google ou Yahoo ne proposent pas de références à partir de ces éléments de code, mais avec le développement des technologies du web 2.0, ils le feront un jour ou l'autre. Pour l'instant, dans le cas où une page contient des liens à l'intérieur d'un menu écrit à l'aide de JavaScript, ces adresses ne seront pas référencées par les robots. Il est donc fortement recommandé d'ajouter les liens en HTML à l'aide de la balise `<a href="mapagevisibleparlesmoteurs.html">`. Après avoir détaillé la position des scripts JavaScript, il faut éclaircir la manière de les insérer. Nous avons déjà vu que JavaScript ne nécessitait pas d'environnement particulier. **Il suffit simplement d'une page HTML à l'intérieur de laquelle vous ajoutez des lignes rédigées en JavaScript** entre deux balises. La première balise indique au navigateur le début du script JavaScript et la seconde en indique la fin. Les deux balises à utiliser sont les suivantes :

- Au début du script : `<script language="javascript">`
- À la fin du script : `</script>`

Entre les deux balises, le nombre de lignes de code est illimité.

#### 4. Javascript et HTML

JavaScript et HTML étaient intimement liés l'un à l'autre, le code HTML servant généralement de conteneur au bloc d'instructions JavaScript. Après chargement du code HTML, le navigateur exécute les blocs d'instructions JavaScript et permet ainsi d'enrichir la page de nouvelles fonctionnalités.

Le code JavaScript s'intègre de deux manières avec le code HTML :

- Insertion directe dans la page HTML
- Insertion comme un module externe

#### 4.1. Insertion directe dans la page HTML

Il existe 2 manières d'insérer du code JavaScript dans une page HTML :

- **Insertion pour exécution directe :** Le code s'exécute automatiquement lors du chargement de la page HTML dans le navigateur en même temps que le contenu de la page HTML s'affiche à l'écran. Le code JavaScript est placé dans le corps même de la page HTML, entre les balises <body> ... </body> : Exemple :

```
<html>

<head>

<title>..... </title> </head>

<body>

    <script language="JavaScript">
        alert('bonjour');

    </script>

</body>

</html>
```

- **Insertion et exécution événementielle :**
  - ✓ Le code JavaScript est d'abord lu par le navigateur, stocké en mémoire, pour ne s'exécuter que sur demande, lors du déclenchement d'un **événement (click sur un lien, saisie ,...)**.
  - ✓ Le code JavaScript est placé dans le corps même de la page HTML, entre les balises <head> ..... et ..... </head>.
  - ✓ Le code s'exécutera seulement lors d'un événement généré par intervention de l'utilisateur.
  - ✓ Il faut écrire le code correspondant à cet **événement** dans le corps du document HTML.
  - ✓ **Exemple :**

```

<html>
<head>
<title> Test </title>
<script language="JavaScript">
    function bonjourClasse (){
        alert ('Bonjour classe');
    }
</script>
</head>
<body>
    <button onclick='bonjourClasse()'>Bonjour</button>
</body>
</html>

```

#### 4.2. Insertion par appel de module externe

On peut insérer du code JavaScript en faisant appel à un module externe se trouvant à n'importe quelle adresse (URI) :

```

<script src="URL du module externe">
.....
</script>

```

- Le code JavaScript est Stocké dans un fichier sur le serveur à son adresse d'appel sous forme de TEXTE SIMPLE portant l'extension **.txt** ou **.js**
- Les deux balises de JavaScript doivent être placés entre les Tags **<body> et </body>** dans le cas d'une exécution directe ou entre les Tags **<head> et </head>** de la page HTML pour une exécution différée.
- **Avantage :** Simplifie la maintenance des sites faisant appel à des modules JavaScript communs à plusieurs pages HTML.
- **Inconvénient :** l'appel au code externe génère une requête supplémentaire vers le serveur, et encombre le réseau

**Exemple :**

Fichier **monscript.js**

```
function bonjourClasse (){  
    alert ('Bonjour classe');  
}
```

Fichier **mapage.html**

```
<html>  
<head>  
<title> Ma page </title>  
    <script language="JavaScript" src= 'monscripte.js' > </script>  
</head>  
<body>  
    <button onclick='bonjourClasse()'>Bonjour</button>  
</body>  
</html>
```

## 5. Javascript n'est pas JAVA

Il importe de savoir que Javascript est **totalemtent différent** de Java. Bien que les deux soient utilisés pour créer des pages Web évoluées, bien que les deux reprennent le terme Java (café en américain), nous avons là deux outils informatiques bien différents.

Javascript	Java
Code intégré dans la page Html	Module (applet) distinct de la page Html
Code interprété par le browser au moment de l'exécution	Code source compilé avant son exécution



Codes de programmation simples mais pour des applications limitées	Langage de programmation beaucoup plus complexe mais plus performant
Permet d'accéder aux objets du navigateur	N'accède pas aux objets du navigateur
Confidentialité des codes nulle (code source visible)	Sécurité (code source compilé)

### En résumé :

- Javascript est plus simple à mettre en œuvre car c'est du code que vous ajouterez à votre page écrite en Html avec par exemple un simple éditeur de texte comme Notepad. Java pour sa part, nécessite une compilation préalable de votre code.
- Le champ d'application de Javascript est toute assez limité alors qu'en Java vous pourrez en principe tout faire.
- Comme votre code Javascript est inclus dans votre page Html, celui-ci est visible et peut être copié par tout le monde (view source). Ce qui pour les entreprises (et les paranoïaques) est assez pénalisant. Par contre, en Java, votre code source est broyé par le compilateur et est ainsi indéchiffrable.