Javascript : Introduction

Contenu

[Présentation 2](#_Toc510001378)

[Un peu d’histoire 2](#_Toc510001379)

[Intégration d’un script Javascript 3](#_Toc510001380)

[Définir le Javascript directement dans la balise HTML 3](#_Toc510001381)

[Placer le code dans une balise <script> 3](#_Toc510001382)

[Définir un script dans un fichier séparé 4](#_Toc510001383)

[Syntaxe 5](#_Toc510001384)

[Bases syntaxiques 5](#_Toc510001385)

[Mots réservés 5](#_Toc510001386)

[Sensibilité à la casse 5](#_Toc510001387)

[Commentaires 5](#_Toc510001388)

[Exercices 7](#_Toc510001389)

# Présentation

JavaScript est un langage de scripts intégré aux pages web (HTML) qui permet d'améliorer la présentation et surtout l'interactivité des pages web.

JavaScript rend les pages web « dynamiques » en manipulant le Document Object Model (D.O.M.), c’est-à-dire la représentation objet de la page web (le « document »). Le Javascript est en effet un langage orienté objet et possède des objets dits natifs (par exemple les objets Window, Date, XmlHttpRequest).

Possibilités offertes par le Javascript :

* Animer du texte et des images
* Ajouter des conditions, des boucles, effectuer des calculs
* Intercepter les événements (souris, clavier …)
* Contrôler les formulaires
* Modifier les caractéristiques des éléments HTML et CSS
* Gérer des menus
* Détecter le navigateur
* Aller chercher des données de façon asynchrone (avec Ajax et l’objet XmlHttpRequest).

Aujourd’hui, il existe des librairies et frameworks en Javascript aussi nombreux que perfectionnés tels que Jquery, AngularJS, VueJS, ReactJS, MeteorJS ou encore NodeJS, une application Javascript côté serveur.

**JavaScript est un langage sensible à la casse. Il est donc obligatoire de respecter les majuscules et les minuscules dans le code ou les instructions (évènements, fonctions natives).**

## Un peu d’histoire

Au milieu des années 1990, la société Netscape développe pour son navigateur web homonyme un langage nommé LiveScript, lequel est repris par la société Sun (rachetée depuis par Oracle)à l’origine de Java, qui lui donne le nom de Javascript, mais il faut bien retenir que, contrairement à ce que l’on peut lire parfois, ces **2 langages sont totalement différents**, leurs seuls points communs étant leur nom et leur lien avec la société Sun.

De son côté, Microsoft créée un langage similaire pour son navigateur Internet Explorer, le JScript, abandonné quelques années plus tard.

Aujourd’hui, le Javascript est standardisé sous la norme ECMA (*European Computer Manufacturers Association*) qui en est à la version ES6 (rédaction de ce document : mars 2018) également connue sous le nom ECMAScript2015 et qui est implémentée par tous les navigateurs récents actuels.

# Intégration d’un script Javascript

Il existe plusieurs méthodes pour intégrer du JavaScript à une page web. Notez que ces méthodes sont similaires à celles vues pour les feuilles de styles CSS.

## Définir le Javascript directement dans la balise HTML

L’exécution de code Javascript peut se faire directement dans une balise HTML via une instruction nommée « événement » : onClick, onMouseOver etc… (nous reviendrons de façon détaillée sur les évènements). Toutefois, c’est désormais **non recommandé**, la bonne pratique étant de mettre tous les scripts dans des fichiers externes.

**En HTML 5, cette pratique est à proscrire car non valide (conformité à la norme W3C HTML 5).**

Exemple :

<!-- Code HTML -->

<html>

<head>

<title>Titre du fichier</title>

</head>

<body>

<a href="page.html" title="Voir la page" onclick="alert("Lien cliqué !");"></a>

</body>

</html>

## Placer le code dans une balise <script>

L’intégration de code Javascript peut également être placé n’importe où dans la page web via une balise <script> :

<!-- Code HTML -->

<html>

<head>

<title>Titre du fichier</title>

</head>

<body>

<a href="page.html" title="Voir la page"></a>

<script>alert("Hello l’AFPA !");"</script>

</body>

</html>

## 

## Définir un script dans un fichier séparé

On peut définir les scripts dans un fichier séparé portant l’extension *.js* qu'il faudra appeler dans les fichiers HTML souhaités. Toute mise à jour du fichier .js ira se répercuter sur toutes les pages HTML utilisant ce script.

**La bonne pratique recommandée aujourd’hui sur le web est d’utiliser les appels de scripts externes. Les appels se font en fin de page, avant la fermeture de la balise </body>.**

L’appel du fichier .js s’effectue à l’aide de la balise <script> et de l’attribut src :

<!-- Code HTML -->

<html>

<head>

<title>Titre du fichier</title>

</head>

<body>

<a href="page.html" title="Voir la page"></a>

<script src="myscript.js"></script>

</body>

</html>

L’attribut src doit contenir le chemin complet vers le fichier (exemple : assets/js/monscript.js) ou d’une URL (exemple : https://code.jquery.com/jquery-3.3.1.js).

## 

# Syntaxe

## Bases syntaxiques

Pour l'écriture des instructions JavaScript, on utilisera l'alphabet ASCII classique (à 128 caractères) comme en HTML.

Les noms de variables et de fonctions ne peuvent comporter d’espaces, d’accents ni de tirets.

Seul le undercore (\_) est autorisé.

**La bonne pratique est d’utiliser la méthode camelCase.**

Une instruction Javacript est écrite sur une ligne et se termine par un point-virgule :

alert("Hello l’AFPA !");

Enfin, JavaScript ignore les espaces, les tabulations et les sauts de lignes.

Pour déclarer une chaîne de caractères, les guillemets " et l'apostrophe ' peuvent être utilisés à condition de ne pas les mélanger. Si vous souhaitez utiliser des guillemets dans vos chaînes de caractères, tapez \" ou \' pour les différencier vis à vis du compilateur.

## Mots réservés

En javascript comme dans les autres langages, il existe des mots réservés correspondant aux instructions du langage qu’il ne faut donc pas utiliser donc comme nom de variables

* [Liste des mots réservés](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Mots_r%C3%A9serv%C3%A9s)

## Sensibilité à la casse

**JavaScript est un langage sensible à la casse. Il est donc obligatoire de respecter les majuscules et les minuscules dans le code ou les instructions (évènements, fonctions natives).**

## Commentaires

Les commentaires sur une ligne débutent par un double slash :

// Cette ligne est commentée

alert("Hello l’AFPA !");

Pour commenter un bloc de code de plusieurs lignes, les commentaires débutent par /\* et se terminent par \*/ :

/\* Bloc de plusieurs lignes en commentaires

alert("Hello l’AFPA !");

console.log("Ce message ne sera pas affiché car commenté dans le code");

\*/

# Exercices

Le but est d’externaliser le code JavaScript dans un fichier .js puis de tester l’intégration de code HTML dans les fonctions JavaScript.

1. **Copier le code suivant dans votre éditeur de texte préféré.**

<!-— Exercice 1 -->

<html>

<head>

<title>Mon 1er script</title>

<script>

alert("Coucou !");

console.log("Ah que Coucou !");

document.write("Coucou beuh !");

</script>

</head>

<body>

<!—- Ajouter le code Javascript ici pour le titre -->

<p>Rien à ajouter</p>

</body>

</html>

Enregistrer le fichier puis l'ouvrir avec votre navigateur. Observez ce qu’il se passe, y compris dans la console.

1. **Créer un fichier JavaScript nommé « exercice1.js »**
2. **Déplacer le code JavaScript du fichier HTML dans le fichier externe JavaScript et importer ce fichier dans le fichier HTML.**
3. **Répéter le titre de la page dans** <body> **avec l’instruction document.write(). Consultez la** [**documentation**](http://www.w3schools.com/jsref/met_doc_write.asp) **sur le site W3Schools. Ajouter du code HTML dans la chaîne de caractères passée en paramètre.**
4. **Modifier la fonction alert() pour y passer en paramètre également du code HTML. Que remarquez-vous ?**