Johann Pardanaud
Sébastien de la Marck

DÉCOUVREZ LE LANGAGE

JAVASCRIPT

OPENCLASSROOMS

EYROLLES

DÉCOUVREZ LE LANGAGE

JAVASCRIPT

Vous connaissez le HTML et avez toujours rêvé d'améliorer le confort de navigation de vos sites web tout en les rendant plus attrayants pour vos visiteurs ? Ce livre est fait pour vous ! Conçu pour les débutants, il vous apprendra pas à pas la programmation en JavaScript, l'un des langages du Web le plus utilisé au monde.

QU'ALLEZ-VOUS APPRENDRE ?

- Premiers pas en JavaScript
- Les variables et les conditions
- Les boucles et les fonctions
- Les objets et les tableaux
- Déboguer le code
- TP : convertir un nombre en toutes lettres

Modeler les pages web

- Manipuler le code HTML
- Les événements et les formulaires
- Manipuler le CSS
- TP : un formulaire interactif

Les objets et les design patterns

- Les objets et les chaînes de caractères
- Les expressions régulières
- Les données numériques
- La gestion du temps
- Les tableaux et les images
- Les polyfills et les wrappers
- Les closures

L'échange de données avec Ajax

- XMLHttpRequest
- Upload via une iframe
- Dynamic Script Loading
- TP : un système d'autocomplétion

JavaScript et HTML 5

- L'audio et la vidéo
- L'élément Canvas
- L'API File et le drag & drop

À PROPOS DES AUTEURS

Sébastien de la Marck, passionné des technologies du Web et plus particulièrement du JavaScript, est l'auteur de plusieurs programmes développés avec ce langage. Il considère que le JavaScript doit être connu des webmasters, en plus du trio HTML/CSS et PHP.

Johann Pardanaud, étudiant et féru d'informatique, découvre les joies de la programmation à la fin de ses années de collège. Très vite, il se lance dans le développement web et tombe sur son langage de prédilection, le JavaScript, qu'il décide d'enseigner via OpenClassrooms.

L'ESPRIT D'OPENCLASSROOMS

Des cours ouverts, riches et vivants, conçus pour tous les niveaux et accessibles à tous gratuitement sur notre plate-forme d'e-éducation : www.openclassrooms.com. Vous y vivrez une véritable expérience communautaire de l'apprentissage, permettant à chacun d'apprendre avec le soutien et l'aide des autres étudiants sur les forums. Vous profiterez des cours disponibles partout, tout le temps : sur le Web, en PDF, en eBook, en vidéo…

www.editions-eyrolles.com

DÉCOUVREZ LE LANGAGE

JAVASCRIPT

DANS LA MÊME COLLECTION

A. Bacco. – **Développez votre site web avec le framework Symfony3.**
N° 14403, 2016, 536 pages.

M. Chavelli. – **Découvrez le framework PHP Laravel.**
N° 14398, 2016, 336 pages.

R. De Visscher. – **Découvrez le langage Swift.**
N° 14397, 2016, 128 pages.

M. Lorant. – **Développez votre site web avec le framework Django.**
N° 21626, 2015, 285 pages.

E. Lalitte. – **Apprenez le fonctionnement des réseaux TCP/IP.**
N° 21623, 2015, 300 pages.

M. Nebra, M. Schaller. – **Programmez avec le langage C++.**
N° 21622, 2015, 674 pages.

SUR LE MÊME THÈME

P. Martin, J. Pauli, C. Pierre de Geyer, É. Daspet. – **PHP 7 avancé.**
N° 14357, 2016, 732 pages.

R. Goetter. – **CSS 3 Flexbox.**
N° 14363, 2016, 152 pages.

E. Biernat, M. Lutz. – **Data science : fondamentaux et études de cas.**
N° 14243, 2015, 312 pages.

B. Philibert. – **Bootstrap 3 : le framework 100 % web design.**
N° 14132, 2015, 318 pages.

C. Delannoy. – **Le guide complet du langage C.**
N° 14012, 2014, 844 pages.

Johann Pardanaud
Sébastien de la Marck

DÉCOUVREZ LE LANGAGE

JAVASCRIPT

EYROLLES

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

Avant-propos

Depuis sa création, le JavaScript n'a cessé de croître en popularité. Longtemps relégué au simple rang de « langage de scripts basiques », il est devenu ces dernières années un langage absolument incontournable pour toute personne s'intéressant à la création de sites web. Il est désormais entré dans la cour des grands, aux côtés d'HTML, de CSS 3 et de PHP.

Le JavaScript est donc devenu un acteur majeur dans la création de sites web. Il a été popularisé par des sites sociaux comme Facebook ou Twitter, qui en font une utilisation massive afin d'améliorer le confort de navigation des utilisateurs : moins de rechargement de pages, interactivité… Le JavaScript a aussi gagné en popularité grâce aux *frameworks* comme jQuery ou Angular.js. Ces frameworks apportent des fonctionnalités en plus, tout en simplifiant et en accélérant le développement de scripts. Cependant, si l'on veut pouvoir tirer parti de ces framewoks, il faut connaître le langage sur lequel ils s'appuient : le JavaScript.

Beaucoup de développeurs apprennent à utiliser jQuery ou Angular.js sans connaître le JavaScript. C'est comme conduire une voiture sans connaître le Code de la route : gare aux accidents ! Les frameworks permettent de faire de grandes choses, mais pas tout ! C'est pourquoi il faut souvent sortir des sentiers battus et mettre les mains dans le JavaScript.

Dans ce livre, nous mettons un point d'honneur à vous enseigner le JavaScript tel que nous aurions voulu qu'il nous soit enseigné. Apprendre JavaScript n'est pas simple ! Bien qu'inspiré de grands langages comme le C, c'est un langage pourvu d'une logique qui lui est propre et que vous ne retrouverez que dans peu d'autres langages. Mais peu importe, ce langage est formidable et passionnant, et offre des possibilités de moins en moins limitées, tant et si bien qu'il devient utilisable ailleurs que dans les pages HTML ! En effet, le JavaScript peut être employé pour réaliser des extensions pour les navigateurs, des programmes et même de la programmation pour serveurs.

Structure de l'ouvrage

Le plan de ce livre a été conçu pour faciliter votre apprentissage du JavaScript. Voici le chemin que nous allons parcourir :

- **Les bases du JavaScript :** cette partie sera consacrée à l'apprentissage des bases du langage. Nous passerons en revue tout ce qui concerne les particularités du langage et nous reprendrons toutes les notions de programmation afin que les débutants en programmation ne soient pas perdus. À la fin de cette partie, vous serez invités à réaliser une application capable d'afficher un nombre en toutes lettres ; elle réutilisera l'ensemble des notions que vous aurez acquises et mettra votre capacité de réflexion à rude épreuve.
- **Modeler les pages web :** après une première partie consacrée aux bases, vous serez ici plongés dans la modélisation de pages web « dynamiques ». Le dynamisme est une notion importante du JavaScript, c'est cela qui permet à l'utilisateur d'interagir avec votre page. Vous découvrirez aussi comment modifier le contenu d'une page web, créer et manipuler des événements et modifier le style CSS de vos éléments HTML. Comme dans la première partie, un TP est destiné à vous faire travailler sur ce que vous aurez appris au cours de votre lecture.
- **Les objets et les design patterns :** une fois les connaissances nécessaires en termes d'interaction avec une page web et les utilisateurs acquises, vous pourrez alors vous investir dans une utilisation un peu plus avancée du JavaScript. Vous y découvrirez comment créer des objets évolués et quels sont les objets natifs les plus intéressants à manipuler. Vous serez aussi initiés au traitement avancé des chaînes de caractères, des nombres et des tableaux, et vous apprendrez à gérer le temps et les images.
- **L'échange de données avec Ajax :** nous aborderons ici la notion d'échange de données entre un client et un serveur et étudierons les diverses possibilités qui s'offrent à nous afin de faire communiquer nos scripts avec un serveur sans pour autant devoir recharger la page web. Un point sera fait sur les structures de données afin que vous sachiez sous quelle forme transférer vos données et vous découvrirez comment réaliser un système d'upload de fichier. La partie se terminera avec la création d'un système d'autocomplétion, chose qui vous resservira très certainement.
- **JavaScript et HTML 5 :** vous avez maintenant terminé votre apprentissage sur tout ce qui était essentiel au JavaScript, dans cette partie nous irons donc un peu plus loin et explorerons la manière dont le HTML 5 permet au JavaScript d'aller encore plus loin. Nous ferons avant tout un point sur ce qu'apporte le HTML 5 par rapport au JavaScript puis nous nous pencherons sur quelques points essentiels, à savoir : la manipulation des balises `< audio >`, `<video >` et `< canvas >`, la gestion des fichiers et la mise en place d'un système de Drag & Drop.
- **Annexes :** vous trouverez ici de quoi approfondir vos connaissances en termes de débogage de code et vous découvrirez les closures qui, bien qu'un brin complexes, vous apporteront un confort appréciable dans votre manière de développer. L'ultime chapitre vous donnera un récapitulatif de certains points essentiels du JavaScript et vous montrera que ce langage peut se rendre très utile même en dehors des navigateurs web.

Comment lire ce livre ?

Suivez l'ordre des chapitres

Lisez ce livre comme on lit un roman. Il a été conçu pour cela.

Contrairement à beaucoup de livres techniques où il est courant de lire en diagonale et de sauter certains chapitres, il est ici fortement recommandé de suivre l'ordre du livre, à moins que vous ne soyez déjà un peu expérimenté.

Pratiquez en même temps

Pratiquez régulièrement. N'attendez pas d'avoir fini de lire ce livre pour allumer votre ordinateur.

http://odyssey.sdlm.be/javascript.htm

Des exercices interactifs sont proposés à la fin de certains chapitres.

Les compléments web

Pour télécharger le code source des exemples de cet ouvrage, veuillez-vous rendre à cette adresse : *http://www.editions-eyrolles.com/dl/0014399*.

Remerciements

Nous tenons à remercier les personnes qui ont contribué de près ou de loin à l'écriture de ce cours. Sans elles, ce cours aurait eu du mal à voir le jour !

Commençons par trois accros du JavaScript :

Yann Logan – Golmote (*https://openclassrooms.com/membres/golmote-13169*) –, pour ses relectures et ses conseils avisés.

Xavier Montillet – xavierm02 (*https://openclassrooms.com/membres/xavierm02-20626*) –, pour ses remarques sur tout ce qui nous semblait négligeable mais qui importait beaucoup au final !

Benoît Mariat – restimel (*https://openclassrooms.com/membres/restimel-61309*) –, qui a activement participé aux débuts quelque peu chaotiques du cours.

Merci encore à vous trois, on se demande toujours ce qu'on aurait fait sans vous !

S'ensuivent certaines personnes d'OpenClassrooms :

Jonathan Baudoin – John-John (*https://openclassrooms.com/membres/john-john-35734*) –, qui a supporté nos coups de flemme pendant plus d'un an !

Pierre Dubuc – karamilo (*https://openclassrooms.com/membres/karamilo-40796*) –, pour nous avoir lancés dans ce projet immense et nous avoir aidés à mettre en place les premiers chapitres du cours.

Mathieu Nebra – M@teo21 (*https://openclassrooms.com/membres/mateo21*) –, sans qui ce cours n'aurait jamais vu le jour puisqu'il est le créateur du Site du Zéro OpenClassrooms.

Merci aussi à nos familles respectives, qui nous ont encouragés pendant plus d'un an et demi !

Merci à vous tous !

Table des matières

Première partie

Les bases du JavaScript

Comme tout langage de programmation, le JavaScript possède quelques particularités : sa syntaxe, son modèle d'objet, etc. En clair, tout ce qui permet de différencier un langage d'un autre. D'ailleurs, vous découvrirez rapidement que le JavaScript est un langage relativement spécial dans sa manière d'aborder les choses. Cette partie est indispensable pour tout débutant en programmation et même pour ceux qui connaissent déjà un langage de programmation car les différences avec les autres langages sont nombreuses.

1 Introduction au JavaScript

Avant d'entrer directement dans le vif du sujet, ce chapitre va vous apprendre ce qu'est le JavaScript, ce qu'il permet de faire, quand il peut ou doit être utilisé et comment il a considérablement évolué depuis sa création en 1995.

Nous aborderons aussi plusieurs notions de bases telles que les définitions exactes de certains termes.

Qu'est-ce que le JavaScript ?

Le JavaScript est un langage de programmation de scripts orienté objet. Dans cette description un peu barbare se trouvent plusieurs éléments que nous allons décortiquer.

Un langage de programmation

Un langage de programmation est un langage qui permet aux développeurs d'écrire du code source qui sera analysé par l'ordinateur.

Un développeur, ou un programmeur, est une personne qui développe des programmes. Ça peut être un professionnel (un ingénieur, un informaticien ou un analyste programmeur) ou bien un amateur.

Le code source est écrit par le développeur. C'est un ensemble d'actions, appelées « instructions », qui vont permettre de donner des ordres à l'ordinateur afin de faire fonctionner le programme. Le code source est quelque chose de caché, un peu comme un moteur dans une voiture : le moteur est caché, mais il est bien là, et c'est lui qui fait en sorte que la voiture puisse être propulsée. Dans le cas d'un programme, c'est pareil : c'est le code source qui régit le fonctionnement du programme.

En fonction du code source, l'ordinateur exécute différentes actions, comme ouvrir un menu, démarrer une application, effectuer une recherche, etc., soit tout ce que l'ordinateur est capable de faire. Il existe énormément de langages de programmation (un bon nombre d'entre eux sont listés sur la page suivante : *http://fr.wikipedia.org/wiki/Cat%C3%A9gorie%3ALangage_de_programmation*).

Programmer des scripts

Le JavaScript permet de programmer des scripts. Comme nous venons de le voir, un langage de programmation permet d'écrire du code source qui sera analysé par l'ordinateur. Il existe trois manières d'utiliser du code source.

- Langage compilé : le code source est donné à un programme appelé « compilateur » qui va lire le code source et le convertir dans un langage que l'ordinateur sera capable d'interpréter : c'est le langage binaire, fait de 0 et de 1. Les langages comme le C ou le C++ sont des langages dits compilés.
- Langage précompilé : ici, le code source est compilé partiellement, généralement dans un code plus simple à lire pour l'ordinateur, mais qui n'est pas encore du binaire. Ce code intermédiaire devra être lu par ce qu'on appelle une « machine virtuelle » qui exécutera ce code. Les langages comme le C# ou le Java sont dits précompilés.
- Langage interprété : dans ce cas, il n'y a pas de compilation. Le code source reste tel quel, et si l'on veut l'exécuter, on doit le fournir à un interpréteur qui se chargera de le lire et de réaliser les actions demandées. Éventuellement, pour obtenir de significatifs gains de performances, le code peut être compilé à la volée (*http://fr.wikipedia.org/wiki/Compilation_%C3%A0_la_vol%C3%A9e*) pendant son exécution. C'est aujourd'hui ce que font la plupart des interpréteurs JavaScript.

Les scripts sont majoritairement interprétés. Et quand on dit que le JavaScript est un langage de scripts, cela signifie qu'il s'agit d'un langage interprété ! Il est donc nécessaire d'utiliser un interpréteur pour faire fonctionner du code JavaScript, ce que vous faites fréquemment car il est inclus dans votre navigateur web !

En effet, chaque navigateur possède un interpréteur JavaScript propre :

- pour Internet Explorer, il s'agit de *Chakra* (l'interpréteur JavaScript des versions antérieures à IE 9 est *JScript*) ;
- pour Mozilla Firefox, l'interpréteur se nomme *SpiderMonkey ;*
- pour Google Chrome, il s'agit de *V8*.

Un langage orienté objet

Intéressons-nous maintenant à la notion « orienté objet ». Ce concept est assez compliqué à définir à ce stade, il sera approfondi par la suite, notamment dans la deuxième partie du livre. Sachez toutefois qu'un langage de programmation orienté objet est un langage qui contient des éléments, appelés « objets », lesquels possèdent des caractéristiques spécifiques et peuvent être utilisés de différentes manières. Le langage fournit des objets de base comme des images, des dates, des chaînes de caractères, etc., mais

il est également possible de créer soi-même des objets pour se faciliter la vie et obtenir un code source plus clair (donc plus facile à lire) et une manière de programmer beaucoup plus intuitive (et donc plus logique).

Il est probable que vous n'ayez rien compris à ce passage si vous n'avez jamais fait de programmation, mais ne vous en faites pas : vous comprendrez bien assez vite comment tout cela fonctionne.

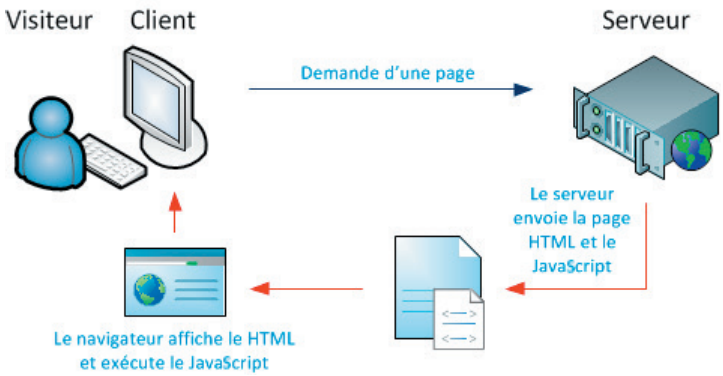
Le JavaScript, le langage de scripts

Le JavaScript est majoritairement utilisé sur Internet, conjointement avec les pages web HTML dans lesquelles il est inclus (ou dans un fichier externe). Le JavaScript permet de « dynamiser » une page HTML en ajoutant des interactions avec l'utilisateur, des animations, de l'aide à la navigation. Par exemple :

- afficher/masquer du texte ;
- faire défiler des images ;
- créer un diaporama avec un aperçu « en grand » des images ;
- créer des infobulles.

Le JavaScript est un langage dit *client-side*, c'est-à-dire que les scripts sont exécutés par le navigateur de l'internaute (le client). Cela diffère des langages de scripts dits *server-side* qui sont exécutés par le serveur web. C'est le cas des langages comme le PHP (*https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql*).

Cette distinction est importante car la finalité des scripts client-side et server-side n'est pas la même. Un script server-side va s'occuper de « créer » la page web qui sera envoyée au navigateur. Ce dernier va alors afficher la page, puis exécuter les scripts client-side tels que le JavaScript. Voici un schéma reprenant ce fonctionnement :



JavaScript est un langage dit client-side,
c'est-à-dire interprété par le client (le navigateur)

Le JavaScript, pas que le Web

Si le langage JavaScript a été conçu pour être utilisé conjointement avec le HTML, il a depuis évolué vers d'autres destinées. En effet, il est régulièrement utilisé pour réaliser des extensions pour différents programmes : Chrome et Firefox possèdent tous deux un panel gigantesque d'extensions en partie codées en JavaScript.

Mais là où le JavaScript a su grandement évoluer ces dernières années, c'est dans la possibilité d'être exécuté sur n'importe quelle machine. Quelques projets permettent d'utiliser le JavaScript en dehors de votre navigateur, le plus connu est sans nul doute Node.js (*https://openclassrooms.com/informatique/cours/des-applications-ultra-rapides-avec-node-js*).

Le JavaScript peut aussi être utilisé pour réaliser des applications. L'interface de Firefox est notamment développée avec ce langage mais cela reste une implémentation bien particulière à la société Mozilla. Cependant, il vous est tout à fait possible aujourd'hui de créer une application en JavaScript grâce à Node.js et si vous souhaitez y ajouter une interface, d'autres projets venant se greffer à Node vous faciliteront la tâche, tel qu'Electron *(http://electron.atom.io/)* ou NW.js (*https://github.com/nwjs/nw.js*).

Petit historique du langage

Brendan Eich, le papa de JavaScript

En 1995, Brendan Eich travaillait chez Netscape Communication Corporation, la société qui éditait le célèbre navigateur Netscape Navigator, alors principal concurrent d'Internet Explorer.

Brendan développe le LiveScript, un langage de scripts qui s'inspire du langage Java, et qui est destiné à être installé sur les serveurs développés par Netscape. Netscape se met à développer une version client du LiveScript, qui sera renommée JavaScript en hommage au langage Java créé par la société Sun Microsystems. En effet, à cette époque, le langage Java était de plus en plus populaire, et appeler le LiveScript « JavaScript » était une manière de faire de la publicité à la fois au Java et au JavaScript lui-même. Mais attention, au final, ces deux langages sont radicalement différents ! N'allez pas confondre le Java et le JavaScript car ils n'ont clairement pas le même fonctionnement.

Le JavaScript sort en décembre 1995 et est embarqué dans le navigateur Netscape 2. Le langage est alors un succès, si bien que Microsoft développe une version semblable, appelée JScript, qu'il embarque dans Internet Explorer 3, en 1996.

Netscape décide d'envoyer sa version de JavaScript à l'ECMA International (*European Computer Manufacturers Association* à l'époque, aujourd'hui *European Association for Standardizing Information and Communication Systems*) pour que le langage soit standardisé, c'est-à-dire pour qu'une référence du langage soit créée et qu'il puisse ainsi être utilisé par d'autres personnes et embarqué dans d'autres logiciels. L'ECMA International standardise le langage sous le nom d'ECMAScript (*http://fr.wikipedia.org/wiki/ECMAScript*).

Depuis, les versions de l'ECMAScript ont évolué. La version la plus connue et mondialement utilisée est la version ECMAScript 5, parue en décembre 2009.

L'ECMAScript et ses dérivés

L'ECMAScript est la référence de base dont découlent des implémentations. On peut évidemment citer le JavaScript, qui est implémenté dans la plupart des navigateurs, mais aussi :

- JScript (*http://fr.wikipedia.org/wiki/JScript*), qui est l'implémentation embarquée dans Internet Explorer. C'est aussi le nom de l'ancien interpréteur d'Internet Explorer ;
- JScript.NET, qui est embarqué dans le framework .NET de Microsoft ;
- ActionScript (*http://fr.wikipedia.org/wiki/ActionScript*), qui est l'implémentation faite par Adobe au sein de Flash ;
- EX4 (*http://fr.wikipedia.org/wiki/E4X*), qui est l'implémentation de la gestion du XML d'ECMAScript au sein de SpiderMonkey, l'interpréteur JavaScript de Firefox.

La plupart de ces implémentations sont quelque peu tombées en désuétude hormis le JavaScript qui a su se frayer une place de choix.

Les versions du JavaScript

Les versions du JavaScript sont basées sur celles de l'ECMAScript (ES). Ainsi, il existe :

- ES 1 et ES 2, qui sont les prémices du langage JavaScript ;
- ES 3 (sorti en décembre 1999) ;
- ES 4, qui a été abandonné en raison de modifications trop importantes qui ne furent pas appréciées ;
- ES 5 (sorti en décembre 2009), la version la plus répandue et utilisée à ce jour ;
- ES 6 finalisé en décembre 2014 et dont l'implémentation avait déjà été commencée avant cette date au sein de plusieurs navigateurs.

Ce cours portera sur la version 5 de l'ECMAScript, la version 6 n'étant pas encore très bien supportée à l'heure où nous écrivons ces lignes.

Un logo inconnu

Il n'y a pas de logo officiel pour représenter le JavaScript. Cependant, le logo suivant est beaucoup utilisé par la communauté, surtout depuis sa présentation à la JSConf EU de 2011. Vous pourrez le trouver à cette adresse : *https://github.com/voodootikigod/logo.js* sous différents formats. N'hésitez pas à en abuser en cas de besoin.

Ce logo non officiel est de plus en plus utilisé.

En résumé

- Le JavaScript est un langage de programmation interprété, c'est-à-dire qu'il a besoin d'un interpréteur pour pouvoir être exécuté.
- Le JavaScript est utilisé majoritairement au sein des pages web mais son utilisation en guise de serveur ou d'application commence à se répandre.
- Tout comme le HTML, le JavaScript est généralement exécuté par le navigateur de l'internaute : on parle d'un comportement client-side, par opposition au server-side lorsque le code est exécuté par le serveur.
- Le JavaScript est standardisé par l'ECMA International sous le nom d'ECMAScript, qui constitue la référence du langage.
- La dernière version standardisée du JavaScript est basée sur l'ECMAScript 5, sorti en 2009. Mais sa nouvelle version, ECMAScript 6, prend du terrain.

QCM
(*http://odyssey.sdlm.be/javascript/01/partie1/chapitre1/qcm.htm*)