

# Javascript, le "vilain petit canard" devenu couteau suisse du Web

LEMONDE.FR | 08.02.12 |



**Limité, lourd, lent, conçu pour les amateurs... A la fin des années 1990, le langage Javascript n'a pas vraiment la cote. Pour de nombreux informaticiens, c'est un outil insuffisant, tout juste bon à [animer](#) des boutons sur des pages Web, mais en aucun cas un "véritable" langage de programmation.**

Quinze ans plus tard, la donne a changé. Javascript est partout : sur les sites Web, dans les applications... Couplé au HTML 5, la toute dernière version du langage à la base des pages Web, il permet de [créer](#) des pages dynamiques, des [jeux parfois très complexes](#), et des lecteurs vidéo qui remplacent peu à peu les anciens lecteurs Flash.

## "UN DÉNOMINATEUR COMMUN"

Que s'est-il passé, en quinze ans, pour que Javascript passe du statut de "vilain petit canard" à celui d'outil incontournable ? Le langage a bien sûr évolué, mais *"si c'est le langage roi aujourd'hui, c'est surtout parce que c'est une plateforme ouverte à tous, un dénominateur commun entre tous les navigateurs"*, juge Lars Bak, responsable du moteur Javascript de Google Chrome, la partie du navigateur chargée d'[accélérer](#) l'exécution du code. *"Pour une entreprise qui se lance aujourd'hui, [utiliser](#) ce langage est une évidence : il suffit de [créer](#) une application une seule fois, et elle fonctionnera partout !"*

Au milieu des années 1990, lorsque [Brendan Eich](#) crée, pour Netscape, la première version de Javascript, l'idée n'est pas de [proposer](#) un standard permettant de [lancer](#) des applications déployables sur smartphones et sur navigateurs : les premiers n'existent pas et concernant les

seconds, Netscape jouit d'une domination encore sans partage. Le Web commence à peine à se [démocratiser](#) et les pages Web se limitent à du texte et des images, mises en page et reliées par des liens. Pour [introduire](#) des fonctionnalités dynamiques – par exemple, [faire évoluer](#) une page en fonction des déplacements de la souris – le HTML ne suffit pas.

Brendan Eich développe alors un langage simplifié, dont les commandes peuvent [être](#) facilement intégrées dans une page Web. Conçu originellement pour [exécuter](#) des tâches assez simples, le langage n'est pas optimisé pour des applications "lourdes". Au début des années 2000, Javascript est de plus en plus utilisé pour des fonctions complexes, mais la lenteur d'exécution des commandes est un frein important. A [partir](#) de 2005, alors que la compétition entre éditeurs de navigateurs Internet prend son envol, la vitesse d'exécution du Javascript sur des pages élaborées – messageries électronique, agrégateurs de flux RSS – devient un enjeu important.

## LA VITESSE DES MOTEURS CONTINUE DE PROGRESSER

*"Lorsque j'ai rejoint Google, en 2006, Javascript était très lent. Et il était très lent sur tous les navigateurs : c'est un langage très dynamique, dont le code peut [changer](#) à la volée, mais en intégrant dans le navigateur un moteur qui facilite son exécution, on peut [obtenir](#) des gains de vitesse très significatifs",* note Lars Bak, responsable du moteur V8 développé par Google pour son navigateur Chrome. Au fil des années, les performances s'envolent : la fondation Mozilla – cofondée par Brendan Eich –, mais aussi Microsoft, Opera, puis Google travaillent d'arrache-pied à [améliorer](#) le rendu du code Javascript dans les navigateurs. Conséquence directe : des applications riches en Javascript, qui auraient été inutilisables dans un navigateur il y a quelques années, font aujourd'hui leur apparition, comme des émulateurs de jeux vidéo. Et la vitesse des moteurs continue de [progresser](#), laissant [entrevoir](#) de nouvelles possibilités.

S'il s'est imposé comme un standard – sous le nom ECMAScript – et un dénominateur commun, Javascript restera-t-il pour autant le principal "couteau suisse" du Web ? Google tente de [populariser](#) sa propre alternative au Javascript, Dart, conçu pour [être](#) plus simple à [programmer](#) et plus rapide. *"Nous avons voulu [offrir](#) une alternative : si vous utilisez Dart sur votre site, les utilisateurs dont le navigateur supporte Dart fonctionnera plus vite. S'il ne le supporte pas, l'utilisateur aura accès à une version Javascript un peu plus lente",* détaille Lars Bak.

Mais pour Brendan Eich, un déploiement à grande échelle de Dart est très peu probable. Le créateur de Javascript estimait, [lors d'une conférence récente](#), que ni Internet Explorer ni Safari ne souhaiteraient [intégrer](#) le nouveau langage. Et même si Dart parvient à s'[imposer](#) progressivement, il continuera de [produire](#) des versions Javascript du code pour que tous les navigateurs puissent [lire](#) les pages correctement. Laisant à Javascript, au moins pour plusieurs années encore, sa place de "dénominateur commun" entre toutes les plateformes.

Damien Leloup