

Améliorations de JavaScript

Dans cette lecture, vous découvrirez l'histoire de JavaScript et l'importance de l'ECMA (European Computer Manufacturers Association) et de l'ECMAScript.

JavaScript est un langage de programmation qui a connu des débuts modestes.

Il a été créé en seulement 10 jours en 1995 par une seule personne, Brendan Eich, qui avait pour mission de créer un langage de script simple à utiliser dans la version 2 du navigateur Netscape. Il s'appelait initialement LiveScript, mais comme le langage Java était très populaire à l'époque, le nom a été changé en JavaScript - bien que Java et JavaScript n'aient rien à voir l'un avec l'autre.

Pendant les premières années qui ont suivi sa création, JavaScript était un simple langage de script permettant d'ajouter des effets de survol de la souris et d'autres formes d'interactivité. Ces effets étaient ajoutés aux pages web à l'aide de l'élément HTML<script>.

À l'intérieur de chaque élément script, il pouvait y avoir du code JavaScript. En raison de la règle selon laquelle HTML, CSS et JavaScript doivent être rétrocompatibles, même le code le plus avancé écrit en JavaScript aujourd'hui finit par être écrit entre ces balises script.

Au fil des ans, JavaScript est devenu de plus en plus puissant et, ces derniers temps, il est continuellement présenté comme l'un des trois langages les plus couramment utilisés.

En 1996, Netscape a conclu un accord avec l'ECMA (European Computer Manufacturers Association) pour rédiger la spécification du langage JavaScript et, en 1997, la première édition de la spécification ECMAScript a été publiée.

L'ECMA publie cette spécification sous la forme de la norme ECMA-262.

Vous pouvez considérer une norme comme une manière convenue de faire fonctionner les choses. Ainsi, l'ECMA-262 est une norme qui spécifie comment le langage JavaScript doit fonctionner.

Il y a eu 12 mises à jour de la norme ECMA-262, la première datant de 1997.

JavaScript en tant que langage n'est pas une entité complètement séparée et autonome. Il n'existe qu'en tant qu'implémentation. Cette implémentation est connue sous le nom de moteur JavaScript.

Traditionnellement, le seul environnement dans lequel il était possible d'exécuter un moteur JavaScript était le navigateur. Plus précisément, un moteur JavaScript n'était qu'un autre élément constitutif du navigateur. Il était là pour aider le navigateur à

atteindre l'objectif de ses utilisateurs, à savoir utiliser l'internet pour le travail, la recherche et les loisirs.

Ainsi, lorsque les développeurs écrivent du code JavaScript, ils l'utilisent pour interagir avec un moteur JavaScript. En d'autres termes, les développeurs écrivent du code JavaScript pour pouvoir "parler" à un moteur JavaScript.

En outre, le moteur JavaScript lui-même est livré avec différents moyens d'interagir avec d'autres parties du navigateur. C'est ce qu'on appelle les API du navigateur.

Ainsi, le code que vous écrivez dans le langage de programmation JavaScript vous permet de.. : 1. Interagir avec le moteur JavaScript à l'intérieur du navigateur 2. Interagir avec d'autres fonctionnalités du navigateur qui existent en dehors du moteur JavaScript, mais qui se trouvent toujours dans le navigateur.

Si, traditionnellement, il était possible d'interagir avec le moteur JavaScript uniquement à l'intérieur du navigateur, tout a changé en 2009, lorsque Node.js a été créé par Ryan Dahl.

Il a trouvé un moyen d'utiliser un moteur JavaScript en tant qu'entité autonome. Soudain, il était possible d'utiliser JavaScript en dehors du navigateur, en tant que programme distinct sur la ligne de commande ou en tant qu'environnement côté serveur.

Aujourd'hui, JavaScript est omniprésent et fonctionne dans les navigateurs, sur les serveurs, en fait, sur tout appareil pouvant exécuter un moteur JavaScript.