

ES6

JavaScript existe depuis longtemps et est resté longuement cantonné à la dynamisation des pages web. Mais depuis quelques années il sort de ce rôle et se déploie avec efficacité dans d'autres contextes. Par exemple son utilisation côté serveur avec [node.js](#) est un franc succès.□

Officiellement JavaScript se nomme [ECMAScript et a une normalisation](#). On trouve les dernières fonctionnalités et définitions dans la norme [Ecma-262](#).

Vous avez un bon résumé de l'historique et de la situation actuelle dans [cet article de Wikipédia](#).

Ce que vous trouvez comme possibilités dans votre navigateur correspond normalement à cette norme avec en plus des objets et méthodes propres au navigateur.

Le souci c'est que les navigateurs ont du mal à suivre !

En gros aujourd'hui **ES5** (ECMAScript Edition 5) est bien pris en charge et ne pose pas de problème. Par contre **ES6** (aussi dénommé ES2015 parce que paru en 2015) [n'est pas encore totalement supporté](#). Mais ce n'est pas une raison pour s'en priver ! La solution consiste à écrire le code en ES6 et à le transformer automatiquement en ES5. On verra dans ce cours comment faire ceci facilement.

Dans ce cours je vous présente **ES6** qui conserve l'existant (donc tout ES5 reste valable) et ajoute de nombreuses et intéressantes possibilités, en particulier les modules et les classes.□

Cette série d'articles n'est pas une initiation à JavaScript mais s'adresse à ceux qui savent déjà coder avec **ES5** et qui désirent aller plus loin avec les nouvelles possibilités d'**ES6**.

Les évolutions

- [Les chaînes de caractères](#)
- [Les variables](#)
- [Les paramètres des fonctions](#)
- [Les fonctions](#)
- [Les objets](#)
- [Les tableaux](#)
- [Les tableaux typés](#)

Les nouveautés

- [Les symboles](#)
- [La déstructuration](#)
- [Les collections avec clés \(set et map\)](#)
- [Boucles, itérations et générateurs](#)
- [Les promesses](#)
- [Les classes](#)
- [Les modules](#)