

# C'est quoi ES6 ?

*ES5, ES6, ECMAScript ? C'est quoi tous ces noms bizarres ? Et quel est le rapport avec Javascript ?*

**Et si on mettait un peu d'ordre dans tout ça... Je peux te garantir qu'à la fin de cette vidéo, tu comprendras tout.**

## Dejà c'est quoi ES ou ECMAScript ?

Prenons le cas d'un fabricant de jouet pour bébés.

S'il veut pouvoir vendre ses jouets au grand public alors ses jouets devront respecter une certaine norme, un certain standard, ISO quelque chose.

Par exemple, les jouets pour bébé n'auront jamais d'angles pointus. Sécurité oblige...

Alors ECMAScript c'est exactement ça, c'est un standard de langage de programmation, il définit:

- La syntaxe
  - Les types de variable
- et encore pas mal d'autres choses...

## Quel est le rapport avec Javascript ?

Tout simplement, Javascript est un des langages qui respecte le standard ECMAScript. C'est tout !

Il y a d'autres langages qui respectent ce standard, comme ActionScript ou JScript.

## Et c'est quoi les chiffres dans ES6, ES7, etc...?

Le standard ECMAScript évolue ! Il n'est pas figé... Ce qui explique les chiffres 5, 6 7, etc..

Il évolue pour que les langages basés dessus comme le Javascript soient de plus en plus puissants: plus simples, plus riches, etc...

A chaque version, on rajoute des fonctionnalités, on simplifie la syntaxe, etc..

## Faisons un peu d'histoire

**ES5:** Il a été publié en 2009

**ES6:** Il a été publié en 2015, d'ailleurs vous pouvez aussi le voir aussi sous le nom ES2015

Il y a eu un saut particulièrement important entre ES5 et ES6.

Beaucoup de choses ont évolué en ES6... Fonctions Fléchées, Classes, Modules et j'en passe... ça a été une vraie révolution du Javascript

**ES7:** Il a été publié en 2016, d'ailleurs vous pouvez aussi le voir aussi sous le nom ES2016

**ES8:** Il a été publié en 2017, d'ailleurs vous pouvez aussi le voir aussi sous le nom ES2017

**ES9:** Il est déjà en cours...

## Notre Navigateur évolue moins vite que les standards ECMAScript

Aujourd'hui tous les navigateurs modernes sont capables de comprendre le Javascript de norme ES5.

Et ils evoluent pour comprendre les normes encore plus avancées. Mais ils ne sont pas encore capables de comprendre 100% des ajouts d'ES6 et encore moins d'ES7.

Si vous souhaitez connaitre l'évolution exacte, c'est-à-dire quel navigateur supporte quelle fonctionnalité, regardez le lien suivant:

<http://kangax.github.io/compat-table>

## Comment développer moderne dès aujourd'hui ?

**Pourquoi le navigateur peut ne pas comprendre notre code ES6, ES7, etc..?**

Il y a principalement 2 raisons:

- 1- Soit on utilise une nouvelle fonctionnalité (nouvelle fonction, nouvel objet) qu'il ne supporte pas encore
- 2- Soit on utilise une syntaxe qu'il n'est pas encore capable de comprendre

Alors pour résoudre ces problemes et développer moderne avec ces nouveaux standards dès aujourd'hui, on utilisera des **polyfills** et des **transpilers**.

## Qu'est ce qu'un polyfill et un transpiler ?

Un polyfill c'est un bout de code qui va ajouter de la fonctionnalité à notre navigateur au cas ou il ne la supporterait pas encore. C'est-à-dire que si notre navigateur web ne dispose pas encore de certaines fonctionnalités d'ES6 ou ES7 alors grâce aux polyfills qu'on ajoutera à notre code, on pourra quand même bénéficier de ces fonctionnalités. Et ni vu ni connu les fonctionnalités ES6, ES7 marcheront à la perfection !

Un transpiler c'est un traducteur de code. On code en ES6 et magie magie en appuyant sur un bouton, tout notre code écrit avec la syntaxe ES6 est traduit en code ES5 équivalent. Et bien sûr comme nos navigateurs comprennent très bien le ES5, tout marchera à la perfection.

Voilà j'espère t'avoir aidé à y voir plus clair, si tu as appris quelquechose de nouveau alors je compte sur toi pour liker et partager à tes amis.

Si tu veux te former au Développement Moderne, je te conseille le cours [Développement Moderne Javascript et ES6,ES7](#)

Si tu souhaites apprendre Javascript de A à Z et devenir un Javascript Ninja alors rejoins-moi, je suis actuellement en train de construire la [Formation Complète Javascript](#).