

# Pesquisa sobre JavaScript e TypeScript

**Maurício Generoso**

No início, o nome inicial do Javascript era Mocha, depois foi alterado para LiveScript e por fim JavaScript. LiveScript foi o nome oficial do primeiro lançamento da linguagem para navegadores Netscape 2.0, em Setembro de 1995. A alteração no nome para JavaScript foi pura jogada de marketing devido a popularidade da linguagem Java. O JavaScript tinha o objetivo de tornar as páginas web mais dinâmicas.

Em 1996 a Microfost criou uma linguagem parecida com o Javascript para ser usada no Internet Explorer 3, com isso a Netscapre decidiu normalizar o Javascript e foi criado a organização ECMA, passando então o Javascript a ter seu nome oficialmente trocado para ECMAScript, no entanto até hoje a linguagem ainda é conhecida como JavaScript.

As versões do JavaScript seguem a linha do tempo a seguir.

- ECMAScript 1 – lançado em 1997, a primeira versão e que foi desenvolvida em apenas 10 dias.
- ECMAScript 2 – lançado em Agosto de 1998, houve mudanças para alinhar com o padrão ISSO/IEC 16262.
- ECMAScript 3 – lançado em Dezembro de 1999. Essa versão foi adicionado recursos como do-while, expressão regulares, tratamento de exceções, e outras coisas.
- ECMAScript 4 – essa versão foi abandonada em Junho de 2008. Devido a tantas anterações, seria difícil a migração de versão, então foi decidido que as alterações incrementais para o ECMAScript 3 seria lançado como ECMAScript 5. Além disso, essa versão deveria fazer menos coisas que a ECMAScript 4 estava fazendo, e as atualizações seriam divididas em ECMAScript 6 e 7.

- ECMAScript 5 – lançado em Dezembro de 2009. Foi adicionado o “Strict mode”, entre outras coisas, essa foi a versão incremental do ECMAScript 3, que foi acordado no ECMAScript 4.
- ECMAScript 6 – lançado em Junho de 2015. É a versão que incluiu diversas novas funcionalidades ao JavaScript e será visto maiores detalhes mais adiante.
- ECMAScript 7 – lançado em Junho de 2016.

O TypeScript é uma linguagem de programação criada pela Microsoft lançada em Outubro de 2012. O TypeScript permite realizar outras funcionalidades além das já existentes no ECMAScript, e será visto os principais recursos adicionais que a linguagem adiciona ao ES6 mais adiante.

A versão do ECMAScript 6 foi um divisor de águas na forma de programar o javascript, a popularização do JavaScript cresceu muito deste então. A seguir, é destacado as principais mudanças da versão do ECMAScript 5 para o ECMAScript 6:

- 1 Suporte para constantes;
- 2 Bloco de escopo;
- 3 Função Lambda / Arrow Function;
- 4 Rest parameters;
- 5 Template string;
- 6 Propriedades aprimoradas do objeto;
- 7 Atribuição de desestruturação;
- 8 Módulos;
- 9 Classes;
- 10 Iteradores;
- 11 Funções Geradoras;
- 12 Coleções (Map, Set);
- 13 Novos métodos contruídos para várias classes;
- 14 Promises.

Funcionalidades que permitem um código mais limpo, mais compreensível e maior produtividade para programar, estão entre as listadas como principais diferenças entre as versões. Além disso o TypeScript, que compila o código para JavaScript,

adicionou ainda outras funcionalidades ao ECMAScript 6, conforme pode ser listado abaixo:

- 1 Possibilidade de tipagem de dados;
- 2 Modificadores de acesso aos atributos e métodos de classes;
- 3 Interfaces
- 4 Decorators

Foi desenvolvido exemplos de todos os itens listados entre as diferenças do ECMAScript 5 para o ECMAScript 6 anteriormente, e também exemplos das funcionalidades adicionadas através do TypeScript, todos os exemplos estão disponíveis no Github através do endereço: <https://github.com/programmerGM/es5-es6>.

## Referências

[https://www.youtube.com/playlist?list=PLWhiA\\_CuQkbCX9nHuk4rolDYxlLUwXpNI](https://www.youtube.com/playlist?list=PLWhiA_CuQkbCX9nHuk4rolDYxlLUwXpNI)

[https://developer.mozilla.org/pt-](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Functions/rest_parameters)

[BR/docs/Web/JavaScript/Reference/Functions/rest\\_parameters](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Functions/rest_parameters)

[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment)

[US/docs/Web/JavaScript/Reference/Operators/Destructuring\\_assignment](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment)

<https://codeburst.io/javascript-es6-iterables-and-iterators-de18b54f4d4>

<https://medium.com/emanuelg-blog/4-novos-m%C3%A9todos-de-string-em-es6-que-voc%C3%AAs-deveria-conhecer-e85c7262123c>

<https://take.net/blog/4devs/por-que-usar-typescript/>

<https://www.webdevbr.com.br/tipos-basicos-manual-do-typescript>

<https://take.net/blog/4devs/por-que-usar-typescript/>

<https://imasters.com.br/front-end/javascript-20-anos-de-historia-e-construcao-da-web>

<http://shipit.resultadosdigitais.com.br/blog/javascript-1-uma-breve-historia-da-linguagem/>

<https://www.cleverism.com/skills-and-tools/typescript>