

JavaScript/ECMAScript

JavaScript foi inventado por Brendan Eich em 1995.

Foi desenvolvido para Netscape 2 e tornou-se o padrão ECMA-262 em 1997.

Depois que a Netscape entregou o JavaScript para a ECMA, a fundação Mozilla continuou a desenvolver JavaScript para o navegador Firefox. A última versão do Mozilla foi 1.8.5. (Idêntico ao ES5).

O Internet Explorer (IE4) foi o primeiro navegador a oferecer suporte ao ECMA-262 Edição 1 (ES1).

O Comitê Técnico da ECMA 39

Em 1996, Netscape e Brendan Eich levaram JavaScript para a organização de padrões internacionais ECMA, e um comitê técnico (TC39) foi criado para desenvolver a linguagem.

ECMA-262 Edição 1 foi lançada em junho de 1997.

De ES4 a ES6

O comitê TC39 se reuniu em Oslo em 2008, para trabalhar com o ECMAScript 4 após algumas mudanças o ECMAScript 4 foi renomeado para ES5

O ES5 deve ser uma atualização incremental do ECMAScript 3.

Os recursos do ECMAScript 4 devem ser adquiridos em versões posteriores.

O TC39 deve desenvolver uma nova versão principal, maior em escopo que o ES5.

ES5 foi um grande sucesso. Foi lançado em 2009 e todos os principais navegadores (incluindo o Internet Explorer) eram totalmente compatíveis em julho de 2013:

Chrome 23	IE10 / Edge	Firefox 21	Safari 6	Opera 15
Nov 2012	Sep 2012	May 2013	Jul 2012	Jul 2013

ES6 também foi um grande sucesso. Foi lançado em 2015 e todos os principais navegadores estavam em total conformidade em março de 2017:

Chrome 51	Edge 14	Firefox 52	Safari 10	Opera 38
May 2016	Aug 2016	Mar 2017	Sep 2016	Jun 2016

Aplicações

Front end

Manipulação do DOM, comunicação assíncrona com o back end e mais

Back end

Comunicação com banco de dados, manipulação de arquivos e mais (Node)

Jogos

Geralmente se aproveitando de benefícios do HTML5

Sintaxe Básica

Exemplo 1

```
1  const nome = "Joãozinho";
2
3  function somar(a, b) {
4      return a + b;
5  }
6
7  somar(2, 4);
8  // resultado 6
```

Exemplo 2

```
1  const btn = document.querySelector('.btn');
2
3  btn.addEventListener('click', function() {
```

```
4     btn.classList.add('ativo');
5  });
```

Variáveis

Antes de Iniciarmos

- 1. Criar um documento HTML (index.html)
- 2. Criar um documento JavaScript (script.js)
- 3. Linkar o script ao html

```
<script src="script.js"></script>
```

Variáveis

Responsáveis por guardar dados na memória. Inicia com a palavra `var` , `let` ou `const`

Variáveis	
1	<code>var nome = 'Joãozinho';</code>
2	<code>let idade = 28;</code>
3	<code>const frequente = true;</code>

Evitam repetições

DRY (Don't repeat yourself)

```
1  var preco = 20;
2  var totalComprado = 5;
3  var precoTotal = preco * totalComprado;
```

Sintaxe

Palavra chave `var` seguida do nome, sinal de igual e o valor.

exemplo	
1	<code>var nome = 'Joãozinho';</code>
2	<code>var idade = 28;</code>
3	<code>var frequente = true;</code>

Vírgula

Utilize a vírgula para criar mais de uma variável, sem repetir a palavra chave var.

Vírgula	
1	<code>var nome = 'Joãozinho', idade = 28,</code>
2	<code>frequente = true;</code>

Sem valor

Pode declarar ela e não atribuir valor inicialmente.

Sem valor	
1	<code>var precoAplicativo;</code>
2	<code>// retorna undefined</code>

Nome

Os nomes podem iniciar com `$`, `_`, ou `letras`.

Podem conter números mas não iniciar com eles

Case sensitive

nome é diferente de Nome

Não utilizar palavras reservadas

 :bulb: https://www.w3schools.com/js/js_reserved.asp

Camel case

É comum nomearmos assim: `abrirModal`

Nome

```
1 // Inválido var $nome; var function;
2 var 1frequente;
3
4 // Válido
5 var $selecionar;
6 var _nome;
7 var frequenteNoExterior;
```

Erro ao tentar utilizar uma variável que não foi declarada.

Erro declaração

```
1 console.log(nome);
2 // retorna nome is not defined
```

Hoisting

A declaração das variáveis são movidas para cima do código, porém o valor atribuído não é movido.

Hoisting

```
1 console.log(nome);
2 var nome = 'Joãozinho';
3 // Retorna undefined
4
5 var profissao = 'Programador'; console.log(profissao);
6 // Retorna Programador
```

Mudar o valor atribuído

É possível mudar os valores atribuídos a variáveis declaradas com `var` e `let`. Porém não é possível modificar valores das declaradas com `const`

Valor do atributo

```
1 var idade = 28; idade = 29;
2
3 let preco = 50; preco = 25;
4
5 const frequente = true; frequente = false;
6 // Retorna um erro
```

Exercícios

- Declarar uma variável e atribuir o seu nome
- Declarar uma variável e atribuir sua idade
- Declarar uma variável e atribuir sua comida favorita
- Declarar 3 variáveis sem valores