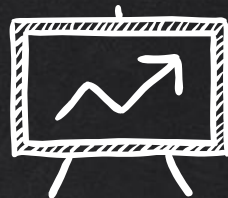


Instituto Federal de São Paulo

Análise e Desenvolvimento de Sistema

EDUCAÇÃO
PÚBLICA
100%
GRATUITA



DESENVOLVIMENTO

WEB II JS

PROFESSOR JOHNATA SANTICIOLI





SUMÁRIOS

- X Document Object Model (DOM)
- X ECMAScript
 - Definição
 - Evolução
- X Hoisting
- X Controle e manipulação de escopo



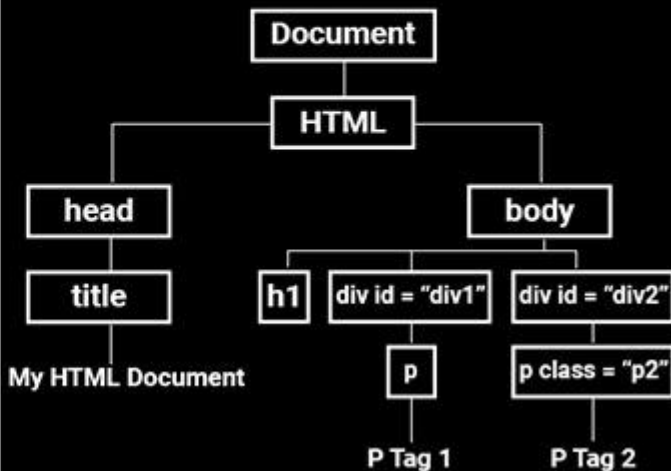


What is Document Object Model ?

HTML Document

```
index.html x
1 <html>
2   <head>
3     <title>My HTML Document</title>
4   </head>
5
6   <body>
7     <h1>Heading</h1>
8     <div id="div1">
9       <p>P Tag 1</p>
10    </div>
11    <div id="div2">
12      <p class="p2">P Tag 2</p>
13    </div>
14  </body>
15 </html>
```

Document Object Model (DOM)





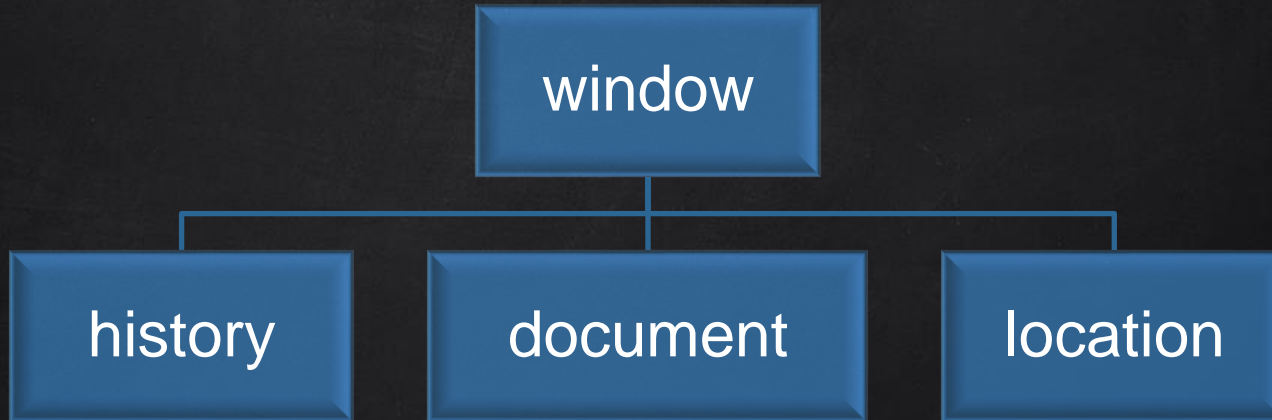
DOM – DOCUMENT OBJECT MODEL

- X DOM – Document Object Model (Modelo Objeto de Documento).
- X Os documentos HTML, assim como os seus elementos, são representados por objetos no JavaScript.
- X Podemos acessar o objeto que representa o documento HTML atual pela constante **document**
 - Todos os demais objetos podem ser acessados através dessa variável





DOM – DOCUMENT OBJECT MODEL





DOM – DOCUMENT OBJECT MODEL

X Métodos para recuperar elementos no HTML pelo:

- id – `document.getElementById('idElement')`
- classe – `document.getElementsByClassName('classElement')`
- tag – `document.getElementsByTagName('tagElement')`
- atributo name – `document.getElementsByName('nameValeu')`
- Seletor CSS:
 - elemento único – `document.querySelector(selectorCSS')`
 - Conjunto de elementos – `document.querySelectorAll(selectorCSS')`

X Exemplo:

- domSeletores.html



REFERÊNCIAS

Você poderá estudar mais sobre o tema nas seguintes páginas:

<https://tableless.com.br/entendendo-o-dom-document-object-model/>

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model





ECMAScript e JavaScript

X **ECMA** – Associação Europeia de Padronização

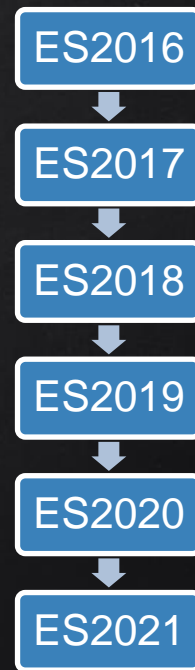
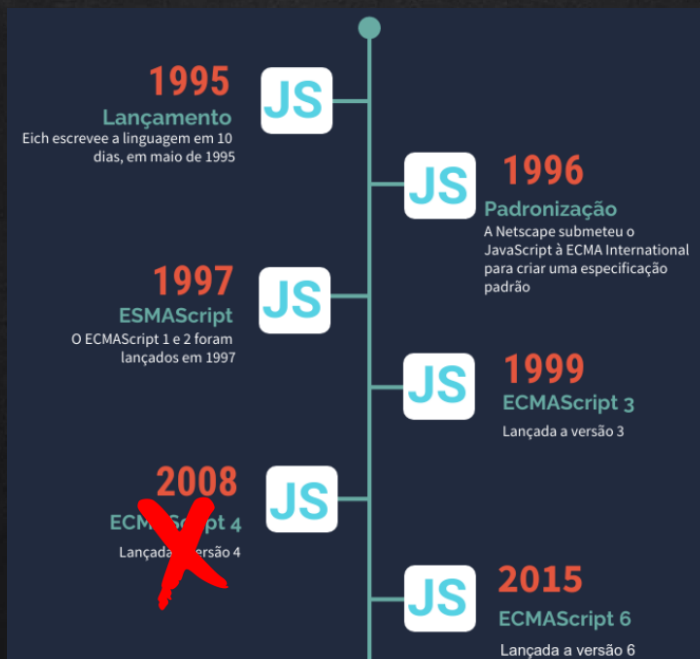
X **ECMAScript** é a padronização do Javascript

- Início **ES1** – JavaScript 1.1
- Atual **ES2021** – 2021/06





LINHA DO TEMPO



ECMA-262



ECMAScript EVOLUÇÃO

- X **ES1** – Padronização Inicial JS 1.1
- X **ES2** – Pequenas atualizações e correções de *bugs*
- X **ES3** – Expressões regulares e tratamento de exceção
- X **ES4** – Versão abandonada
- X **ES5** – JSON e métodos para array (map, filter, reduce ...)
- X **ES6** – (*) String, let, const, desestruturação
- X **ES2016** – ** e Array.includes
- X **ES2017** – Async Functions e métodos novos para objetos
- X **ES2018** – Template string, promisses e métodos novos para expressão regular
- X **ES2019** – ToString e JSON superset
- X **ES2020** – Dynamic imports e string matchAll
- X **ES2021** – [String replaceAll](#) e [Logical OR assignment](#)



REFERÊNCIAS

Você poderá estudar mais sobre o tema nas seguintes páginas:

<https://blog.betrybe.com/javascript/>

<https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>

https://exploringjs.com/impatient-js/ch_new-javascript-features.html

<https://h3manth.com/ES2021/#logical-assignment-operators>





HOISTING

X Hoisting

- Quando o código é executado a primeira vez (parse), o JavaScript eleva declarações de variáveis com var e declaração de funções
- Atenção let e const não são hoisting

X Exemplo:
○ hoisting.js





ESCOPO DE VARIÁVEIS E CONSTANTES

X Declarando variáveis e constantes:

- `var x = 5` – usada para declarar tanto variáveis locais em funções como variáveis globais
- `let x = 5` – usada para declarar uma variável local de escopo de bloco
- `const x = 5` – usada para declarar uma constante local de escopo de bloco





ESCOPO DE VARIÁVEIS E CONSTANTES

X Declarando variáveis e constantes:

- `var x = 5` – sofre hoisting
- O que ocorre se duas ou mais libs(bibliotecas) utilizarem o mesmo nome para variáveis?
- `var` é legado, por isso utilizar somente `const` e `let`

X Exemplo :
○ `variaveis.html`





ESCOPO DE VARIÁVEIS E CONSTANTES

X Declarando variáveis e constantes:

- `let x = 5` e `const x = 5`
- `let` e `const` respeita todos os blocos em função, estruturas de repetição e condição
- `let` = let me change = pode ter seu valor alterado
- `const` = constant= não pode ter ser reatrinuído





ESCOPO DE VARIÁVEIS E CONSTANTES

	global scoped	function scoped	block scoped	reassignable	redeclarable	can be hoisted
var	+	+	-	+	+	+
let	-	+	+	+	-	-
const	-	+	+	-	-	-



REFERÊNCIAS

Você poderá estudar mais sobre o tema nas seguintes páginas:

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/let>

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/const>

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/var>

<https://developer.mozilla.org/pt-BR/docs/Glossary/Hoisting>





THANKS!

Até a próxima aula!!!

www.ifsp.edu.br
johnata.santicioli@ifsp.edu.br

