

# Índice

<a href="#">JavaScript</a>	1
<a href="#">DOM</a>	1
<a href="#">Início</a>	4
<a href="#">Declarando variáveis</a>	4
<a href="#">Código JS:</a>	5
<a href="#">Eventos</a>	6
<a href="#">Formatura</a>	9
<a href="#">Tema</a>	9
<a href="#">Mestre de cerimônia</a>	9
<a href="#">Orador</a>	9
<a href="#">Homenageado</a>	9
<a href="#">Vídeo</a>	9
<a href="#">Links e afins</a>	10
<a href="#">Dúvidas</a>	10

---

## Assuntos: JavaScript.

---

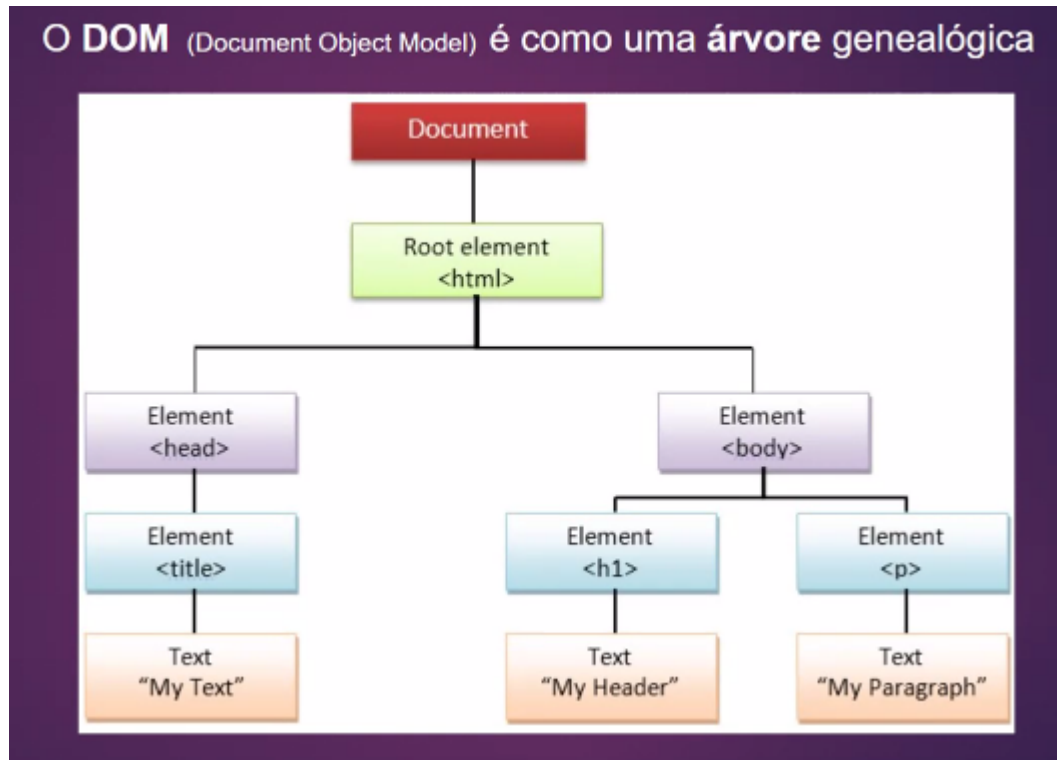
### Abertura Alunos:

Apresentação pitch  
Fulano e Fulano.

---

# JavaScript

## DOM



O Document Object Model (DOM) é uma interface de programação para documentos HTML, XML e SVG. Ele fornece uma representação estruturada do documento, permitindo que programas manipulem a estrutura, o estilo e o conteúdo do documento de forma dinâmica.

O DOM representa o documento como uma árvore de objetos, onde cada nó na árvore representa uma parte do documento (como elementos HTML, atributos, texto etc.). Os programadores podem acessar e manipular esses nós usando métodos e propriedades fornecidos pelo DOM, permitindo a criação de scripts que interagem com o conteúdo HTML de uma página da web de maneira dinâmica.

Por exemplo, um script JavaScript pode usar o DOM para selecionar um elemento específico em uma página da web e alterar seu conteúdo, estilo ou posição. Isso permite a criação de páginas da web interativas e dinâmicas, onde o conteúdo pode ser atualizado e modificado em resposta a eventos do usuário ou ações do script.

Em resumo, o DOM é uma parte fundamental do desenvolvimento da web moderna, permitindo a criação de páginas da web dinâmicas e interativas através da manipulação programática da estrutura e do conteúdo do documento HTML.

**JavaScript é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web.**

### **Tipagem Fraca:**

A tipagem fraca, também conhecida como tipagem dinâmica ou tipagem fraca, refere-se a um sistema de tipos em linguagens de programação onde as variáveis podem ser implicitamente convertidas de um tipo para outro durante a execução do programa. Isso significa que você pode realizar operações com diferentes tipos de dados sem necessariamente converter explicitamente esses tipos.

Por exemplo, em linguagens com tipagem fraca, você pode somar um número inteiro com uma string sem receber um erro. O sistema de tipos fará uma conversão automática para que a operação seja realizada. No entanto, isso também pode levar a resultados inesperados se não for cuidadosamente gerenciado.

### **Multiparadigma:**

Linguagens de programação multiparadigma são aquelas que suportam mais de um paradigma de programação. Um paradigma de programação é uma abordagem ou estilo para a construção de software. Exemplos de paradigmas de programação incluem programação orientada a objetos, programação funcional, programação procedural, entre outros.

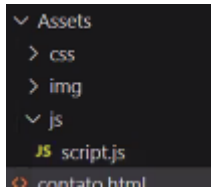
Uma linguagem multiparadigma pode oferecer suporte a diferentes estilos de programação, permitindo que os desenvolvedores escolham a abordagem que melhor se adapta ao problema em questão. Por exemplo, uma linguagem multiparadigma pode permitir que você escreva código orientado a objetos em uma parte do seu programa e código funcional em outra parte.

Exemplos de linguagens de programação multiparadigma incluem Python, JavaScript, Scala e C++.

Em resumo, a tipagem fraca refere-se à flexibilidade do sistema de tipos de uma linguagem, permitindo conversões implícitas entre tipos de dados, enquanto o multiparadigma refere-se à capacidade de uma linguagem suportar diferentes estilos de programação, como orientação a objetos, funcionalidade e procedimental.

# Início

Em assets, crie uma pasta de nome “js”, e crie um arquivo de nome padrão “script.js” para poder fazer o script em Javascript.



Fazendo contato entre as páginas entre HTML e JS, usa a Tag script . Puxamos a tag script = src no final da página, porque diferente do CSS ele primeiro tem que identificar o que foi programado, tem eventos que acionam JS.

## Declarando variáveis

- **let:** A palavra-chave let foi introduzida no ECMAScript 6 (ES6) e tem escopo de bloco.  
Variáveis declaradas com let são acessíveis apenas dentro do bloco em que foram declaradas (dentro de chaves {}).
- **var:** A palavra-chave var era a forma tradicional de declarar variáveis em JavaScript antes das versões mais recentes da linguagem.  
Variáveis declaradas com var têm escopo de função. Isso significa que elas são acessíveis apenas dentro da função em que foram declaradas.
- **const:** A palavra-chave const também foi introduzida no ECMAScript 6 (ES6) e é usada para declarar constantes (que não mude), ou seja, variáveis que não podem ser reatribuídas após a inicialização.  
Assim como let, as variáveis declaradas com const têm escopo de bloco.

Como JS tem uma dinâmica fraca por isso ele automaticamente entende e se adapta ao tipo de dado adicionado, por exemplo número pode ser float, double, etc.... depende de como você conduz o script

### Acessando o DOM:

Devemos buscar informações no DOM, para isso, usamos **window.document.algo**, esse algo é substituível por alguma variável de sua necessidade.

## Código JS:

JS lê o código da direita para esquerda, boa prática também é declarar as variáveis globais no começo do código.

**document.getElementById("nome")** é usado para selecionar um elemento HTML por seu ID.

**document.querySelector("id")** é uma maneira poderosa de selecionar elementos HTML em JavaScript, permitindo que você selecione elementos usando seletores CSS. Quando você usa `querySelector()`, pode especificar um seletor CSS como argumento para selecionar o elemento desejado.

Para selecionar um elemento por seu ID usando `querySelector()`, você pode passar um seletor CSS que começa com # seguido pelo ID do elemento.

`querySelector()` é útil quando você precisa selecionar elementos usando seletores CSS mais complexos do que apenas IDs. Ele permite que você selecione elementos com base em qualquer seletor CSS, como classes, tipos de elemento, atributos, etc.

**innerHTML** a propriedade `innerHTML` permite obter ou definir o conteúdo HTML de um elemento. Ao acessar `innerHTML`, você obtém uma string contendo o HTML dentro do elemento. Ao definir `innerHTML`, você substitui todo o conteúdo HTML dentro do elemento pelo novo conteúdo especificado. Use com cuidado para evitar a perda acidental de conteúdo e considere questões de segurança ao lidar com conteúdo HTML dinâmico.

```
is / js > * scripts > ...  
var nome = document.getElementById("nome");  
var email = document.querySelector("#email"); I
```

### Criando objeto :

**function** declaração de objeto

**ValidarNome ()** nome do objeto

**Let** variável dentro do bloco

**txtNome** nome da variável

**document.getElementById ("txtNome")** caminho no HTML

**If** condição

**Length** tamanho do input

**value** valor do input

**nome** variável do input

**<2** tamanho de caracteres

**innerHTML = "Nome inválido"** : "print in" dentro do input que estamos trabalhando.  
Mensagem exibida caso os parâmetros não sejam atingidos.  
**txtNome.style.color= red**: você consegue manipular a cor do texto.

```
function ValidarNome(){  
    let txtNome = document.getElementById("txtNome");  
    if(nome.value.length < 2){  
        txtNome.innerHTML = "Nome Inválido" | I  
    }  
}
```

## Eventos

Eventos em JavaScript são ações que ocorrem em um documento HTML ou em um elemento específico, como um clique do mouse, pressionamento de tecla, carregamento de página, entre outros. Eles permitem que você crie interações dinâmicas e responsivas em uma página da web.

Eventos são como "funções" que chamam métodos. Ele só chamará o método quando esse evento ocorrer. Criando a função precisa chamar o evento no html dentro do bloco que você deseja.

Evento	Origem	Descrição	Elementos onde se aplica
<b>onload</b>	documento	Disparado quando documento é carregado	body,frameset
<b>onunload</b>	documento	Disparado quando documento é descarregado de janela ou de frame	body,frameset
<b>onsubmit</b>	documento	Disparado quando formulário é submetido	form
<b>onreset</b>	mouse	Disparado quando formulário é "limpado" via botão de reset	form
<b>onselect</b>	mouse	Disparado quando texto é selecionado numa área de entrada de texto	input type="text", ou type="password" ou textarea
<b>onblur</b>	mouse/teclado	Disparado quando elemento de entrada do formulário perde o foco, i.é, qdo texto fora do elemento é selecionado	a, area, button, input, select, textarea

<b>onchange</b>	mouse/teclado	Disparado quando elemento perde o foco e foi modificado	input, select,textarea
<b>onclick</b>	mouse	Disparado quando botão de formulário é pressionado ou link é selecionado via click do mouse	maioria
<b>ondblclick</b>	mouse	Idem, mas click duplo do mouse	maioria
<b>onfocus</b>	mouse/teclado	Disparado quando o elemento recebe foco: clicando o mouse dentro do elemento ou entrando no mesmo via Tab	a, area, button, input, select, textarea
<b>onmousedown</b>	mouse	Disparado quando mouse é pressionado enquanto está sobre um elemento	maioria
<b>onmouseup</b>	mouse	Idem, mas quando mouse é despressionado	maioria
<b>onmousemove</b>	mouse	Disparado quando mouse é movido enquanto sobre elemento	maioria
<b>onmouseout</b>	mouse	Disparado quando mouse é movido fora do elemento onde estava	maioria
<b>onmouseover</b>	mouse	Disparado quando cursor do mouse é movido sobre elemento	maioria
<b>onkeydown</b>	teclado	Disparado quando tecla é pressionada.	maioria
<b>onkeypress</b>	teclado	Disparado quando tecla é pressionada e liberada.	maioria
<b>onkeyup</b>	teclado	Disparado quando tecla pressionada é liberada.	maioria

---

# Formatura

Horário	Descrição	Responsável
16:00 - 16:10	<b>Abertura Sala do Zoom</b> Músicas + Apresentação dos MCs e orientações do Zoom	
16:10 - 16:20	<b>Discurso de Abertura Generation Brasil</b> Convidade faz fala institucional de abertura apresentando quem é a Gen e parabeniza os formades	A confirmar
16:20 - 16:25	<b>Participação Especial</b>	A confirmar
16:25 - 16:40	<b>Discurso dos Participantes</b> Um participante faz discurso sobre a jornada da turma - <b>7.5min para cada turma</b>	T70 T71
16:40 - 16:50	<b>Homenagem</b> Participantes escolhem a quem eles desejam quem homenagear - <b>5min para cada</b>	T70 T71
16:50 - 17:05	<b>Discurso Instrutores Generation Brasil</b> Pessoa instrutora fala sobre a turma e Gen -	T70 T71
17:05 - 17:25	<b>Convocação das Pessoas Participantes</b> Vídeo das pessoas participantes de cada turma - <b>10 min para cada turma</b>	Vídeo T70 Vídeo T71
17:25 - 17:30	<b>Discurso de Encerramento</b> C&I faz declaração de encerramento da turma	A confirmar

## Tema

Matheus Libanio

## Orador

## Homenageado

## Vídeo

sugestões: Vídeo dundler mifflin



---

## Links e afins

**Extensão instalada no VSCode Javascript ES6**

**Apresentação PPTX sobre JavaScript:**

[https://docs.google.com/presentation/d/1ZJXBtqk6rakE\\_ayCbvG\\_ua7Vt6AQgwUc/edit#slide=id.p4](https://docs.google.com/presentation/d/1ZJXBtqk6rakE_ayCbvG_ua7Vt6AQgwUc/edit#slide=id.p4)

**Firefox DevTools doc:**

<https://firefox-source-docs.mozilla.org/devtools-user/>

---

## Dúvidas