

AULA 9 – INTRODUÇÃO AO DOM

- Botão direito no arquivo e após “Open with Live Server”
- O arquivo irá abrir no Chrome, porém, com um micro servidor que ele criou

Estrutura básica do código:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Primeiros passos com DOM</title>
  <style>
    body {
      background-color: rgb(103, 103, 177);
      color: white;
      font: normal 18pt Arial;
    }
  </style>
</head>
<body>
  <h1>Iniciando estudos com DOM</h1>
  <p>Aqui vai o resultado</p>
  <p>Aprendendo a usar o <strong>Dom</strong> em JavaScript</p>
  <div>Clique em mim</div>
  <script>
    //
  </script>
</body>
</html>
```

CSS

HTML

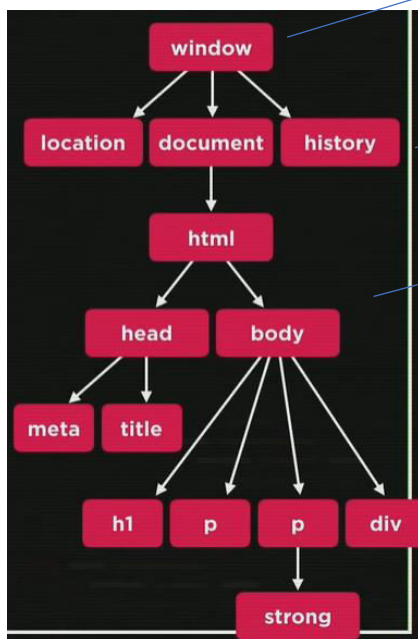
div serve para divisão de documentos em HTML

O QUE É O DOM

→ Document Object Model

→ Basicamente é um conjunto de objetos dentro do navegador que vai dar acesso aos componentes internos do seu website

ÁRVORE DOM



É a raiz da árvore, ou seja, basicamente tudo dentro de JS está dentro de um objeto chamado **window**

Existem muitos outros também, mas esses são os três principais

OBS: tudo o que está abaixo é **child**, e acima é **parent**.
Exemplo: **head** e **body** são **child** de **html**. **html** é **parent** de **head** e **body**, só que é **child** de **document**

- Existem várias maneiras de navegar entre os elementos
- Eu posso **selecionar** esses elementos para navegar dentro da minha Árvore DOM

SELECIONANDO



POR MARCA

→ Está dentro de document

→ **getElementsByTagName()**

→ Com isso, eu consigo selecionar mais de um objeto, até porque existem vários objetos do mesmo tipo, com a mesma tag

Exemplo: tenho duas tags <p> no meu Visual Studio Code

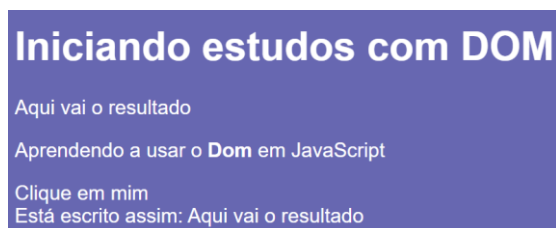
```
<p>Aqui vai o resultado</p>
<p>Aprendendo a usar o <strong>Dom</strong> em JavaScript</p>
```

Código:

```
<script>
  var p1 = window.document.getElementsByTagName('p')[0]
  window.document.write('Está escrito assim: ' + p1.innerText)
</script>
```

Isso significa que o p1 vai receber o que está escrito no primeiro parágrafo
[0] significa 1º parágrafo, [1] significa 2º paragrafo e assim por diante

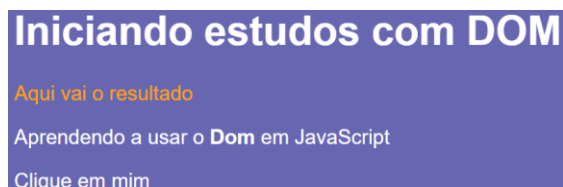
Resultado:



Outro exemplo:

```
<script>
  var p1 = window.document.getElementsByTagName('p')[0]
  p1.style.color = 'orange'
</script>
```

Resultado:



POR ID

→ Identificar um parágrafo

→ **getElementById()**

POR NOME

→ `getElementsByName()`

POR CLASSE

→ `getElementsByClassName()`

OBS:

→ Quando eu tenho o Elements no plural, ou seja, mais de um objeto eu vou ter que achar os **colchetes**

→ Quando eu tenho mais que um objeto, de mesma classe ou mesmo nome, eu vou usar muito o por **MARCA** e também por **NOME**

→ Quando eu quero selecionar apenas um objeto ou família deles, eu utilizo o **ID**

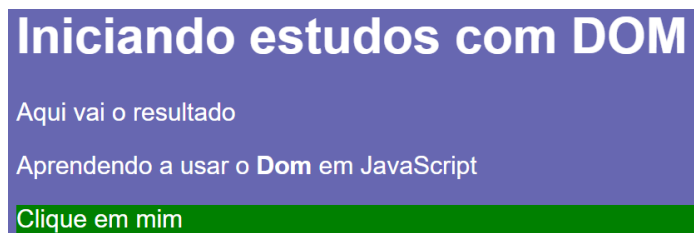
Alguns testes:

Coloco em **div** para assim evitar de ter que criar vários div

```
<div id = 'msg'>Clique em mim</div>
<script>
  var d = window.document.getElementById('msg')
  d.style.background = 'green'
</script>
```

Dentro da minha div, originalmente, está “Clique em mim”. Porém, eu modifiquei utilizando o JavaScript utilizando o DOM a cor de fundo para verde, neste exemplo

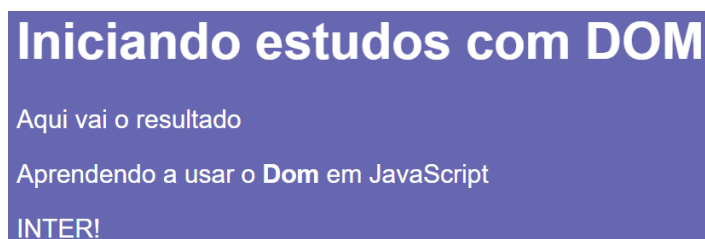
Resultado:



Outro exemplo:

```
<div name = 'msg'>Clique em mim</div>
<script>
  var d = window.document.getElementsByName('msg')[0]
  d.innerHTML = 'INTER!'
</script>
```

Resultado:



Resumo: posso usar qualquer tipo, dependendo da situação, o princípio é o mesmo. Porém, existe uma nova forma:

POR SELETOR

→ `querySelector()`

→ `querySelectorAll()`

Exemplo:

```
<div id = 'msg'>Clique em mim</div>
<script>
  var d = window.document.querySelector('div#msg')
  d.style.color = 'pink'
</script>
```

- Toda DIV é representada pela hashtag #
- Toda CLASSE é representada por um pontinho .

Resultado:

Iniciando estudos com DOM

Aqui vai o resultado

Aprendendo a usar o **Dom** em JavaScript

Clique em mim

Lembrando: se eu utilizar classe tenho que usar o pontinho!

```
<div class = 'msg'>Clique em mim</div>
<script>
  var d = window.document.querySelector('div.msg')
  d.style.color = 'pink'
</script>
```