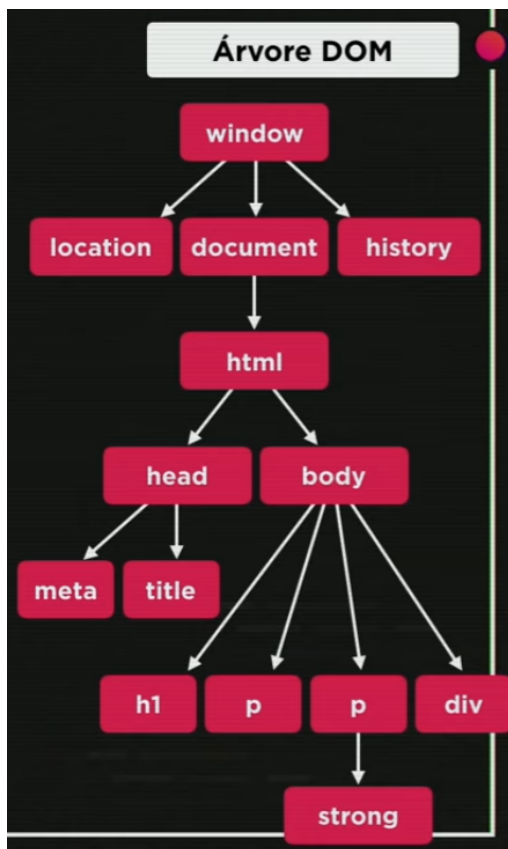


# JAVASCRIPT - MÓDULO 1 - AULA 3

## 1. DOM

O que é o DOM? A sigla dom quer dizer: *Document Object Model* (Modelo de Objeto de Documentos). Basicamente é um conjunto de objetos dentro do seu navegador que vai dar acesso aos componentes internos do seu website.

Vamos entender melhor isso. Basicamente o DOM é semelhante a uma árvore genealógica, só que, a família são os componentes da sua página HTML. Observe a imagem abaixo.



Observe que a árvore DOM é muito semelhante à uma árvore genealógica. E por esse motivo, até as nomeações são parecidas. Por exemplo, na imagem Window está como um “Pai” ou “Mãe” de todos os itens abaixo dele. Quando temos um item que tem itens abaixo dele, chamamos de **PARENT (Pais)**, e os itens que tem um “Pai” são chamados de **CHILD (Filho)**. Como podemos observar então, o document por exemplo é **PARENT** do HTML, e ao mesmo tempo é **CHILD** de Window. Observe também nessa árvore, que todos os itens descendem de Window. Então sabemos sempre que, quando queremos chamar qualquer um dos itens, vamos utilizar a Window como referência. Vamos ver alguns exemplos à frente. Mas antes deles, faça um teste. Abra o VS, crie a estrutura HTML e dentro da tag script escreva **window**. e veja o que o VS vai lhe mostrar de opções.

- 1.1. **window.document:** Esse será talvez a base de comandos mais utilizados em sua codificação JavaScript. Com essa junção de window.document eu estou dizendo ao meu código que eu poderei: Pegar, escrever, alterar, inserir, excluir, etc... Informações, variáveis, arquivos, itens de qualquer tipo ou valor. Basicamente eu posso trabalhar com o que eu precisar dentro de JavaScript. Mas lembrando, isso dentro do documento que eu estou em questão. Vamos ver alguns exemplos.

## 1.2. window.document.write():

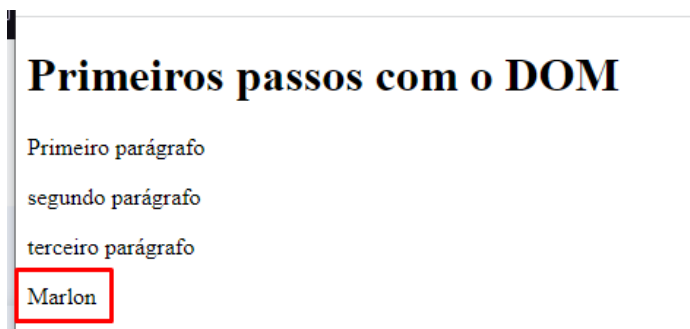
```
<script>
  window.document.write(window.document.URL)
</script>
```

Com esse comando podemos basicamente escrever qualquer item que eu precisar. Seja ele uma URL, texto, variável, informações do seu código e outras diversas possibilidades.

Inicialmente vamos focar o uso do **.write** para transcrevermos os valores, texto e variáveis que estamos aprendendo até o seguinte momento. Para utilizarmos esse comando para escrevermos variáveis por exemplo, basta fazê-lo preenchendo o campo entre parêntesis com a variável selecionada.

```
<script>
  var nome = "Marlon"
  window.document.write(nome)
</script>
```

Portanto isso resultará:



Observe que mesmo tendo **DIVERSAS** opções, o uso do **.write** dentro do JavaScript é bem simples de ser aplicado, basta eu dizer o que quero escrever na tela que ele me traz. Podemos também juntar valores. Texto + variável por exemplo, igual já fizemos com o **window.alert**. Observe:

```
<script>
  var nome = "Marlon"
  window.document.write("Seu nome é: " + nome)
</script>
```

Isso resultará:

## Primeiros passos com o DOM

Primeiro parágrafo

segundo parágrafo

terceiro parágrafo

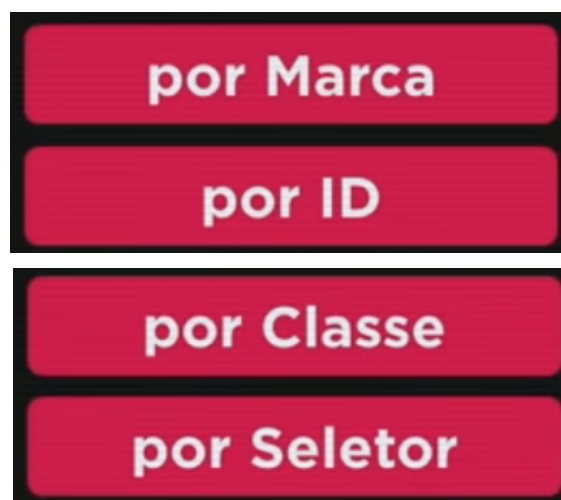
Seu nome é: Marlon

### 1.3. `window.document.getElementById`:

```
<script>  
  window.document.getE  
</script>  
body>  
tml>
```

- `getElement...` (method)
- `getElementsByClassName`
- `getElementsByTagName`
- `getElementsByName`

No tópico anterior vimos a adição de valores com **.write**. Mas como foi citado, o `window.document` tem DIVERSAS maneiras de usabilidade, e uma que você não pode deixar de conhecer é a `.getElementBy...`. Aqui, vamos fazer o seu document ter muito mais conectividade entre HTML e CSS + JavaScript. Pois, ele vai chamar os itens que você quer utilizar do HTML e ou CSS (para n motivos: alterar, excluir, incluir informações, etc) e vai trazer no JavaScript. E para isso temos algumas maneiras de chamar essas TAGs do HTML. Observe.



## EXEMPLOS

- MARCA(TagName):

```
<body>
  <h1>Primeiros passos com o DOM</h1>

  <p>Primeiro parágrafo</p>

  <p>segundo parágrafo</p>

  <p>terceiro parágrafo</p>

  <script>
    var p1 = window.document.getElementsByTagName('p')[0]
    window.document.write(p1.innerHTML)
  </script>
```

SELECIONANDO PELA TAG

PARÁGRAFO

PARÁGRAFO 0

INSERIR O TEXTO DO PRIMEIRO PARÁGRAFO

Resultado:

## Primeiros passos com o DOM

Primeiro parágrafo

segundo parágrafo

terceiro parágrafo

Primeiro parágrafo

- ID:

```
<body>
  <h1>Primeiros passos com o DOM</h1>

  <p id="teste"></p>

  <script>
    document.getElementById("teste").innerHTML = "Hello World";
  </script>
```

ID criado

Adicionando uma informação

Chamando o ID

Resultado:

# Primeiros passos com o DOM

Hello World

- CLASSE(Class Name):

```
<body>
  <h1>Primeiros passos com o DOM</h1>

  <div class="exemplo">Element1</div>
  <div class="exemplo">Element2</div>

  <script>
    var classes = document.getElementsByClassName("exemplo");
    classes[0].innerHTML = "Hello World!";
  </script>
```

DIVs com classe definida

chamando a classe

Especificando a classe e adicionando um texto diferente

Resultado:

# Primeiros passos com o DOM

Hello World! → A que foi alterada!  
Element2

- SELETOR(querySelector)

```
<body>
  <h1>Primeiros passos com o DOM</h1>

  <p>Esse é um parágrafo</p>
  <p id="paragrafo">Esse é um outro parágrafo</p>

  <script>
    document.querySelector("p").style.backgroundColor = "red"
    document.querySelector("#paragrafo").style.backgroundColor = "blue"
  </script>
```

Selecionado apenas pela tag

Selecionado pela TAG + ID

Resultado:

# Primeiros passos com o DOM

Esse é um parágrafo

Esse é um outro parágrafo