

Módulo C: Entendo o DOM

▼ Aula 09) Introdução ao DOM

Pasta de exercícios relacionados: aula09

Extensões para instalar no VS Code:

- Live Server (abre a página HTML que você estiver trabalhando)
- Node.js Exec (habilita a tecla F8 para rodar o JS pelo Node)

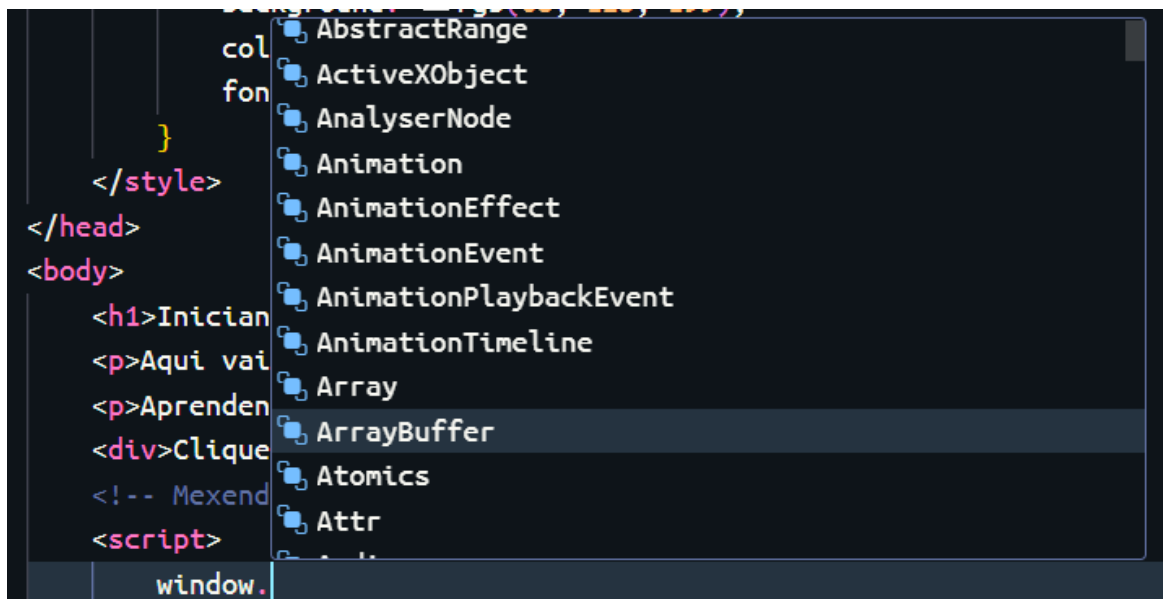
DOM = Document Object Model. É um modelo de objetos para documentos. Dá aos navegadores acesso aos componentes internos do seu site. Não funciona dentro do Node.js.

Há uma árvore DOM, que é formada da seguinte forma:

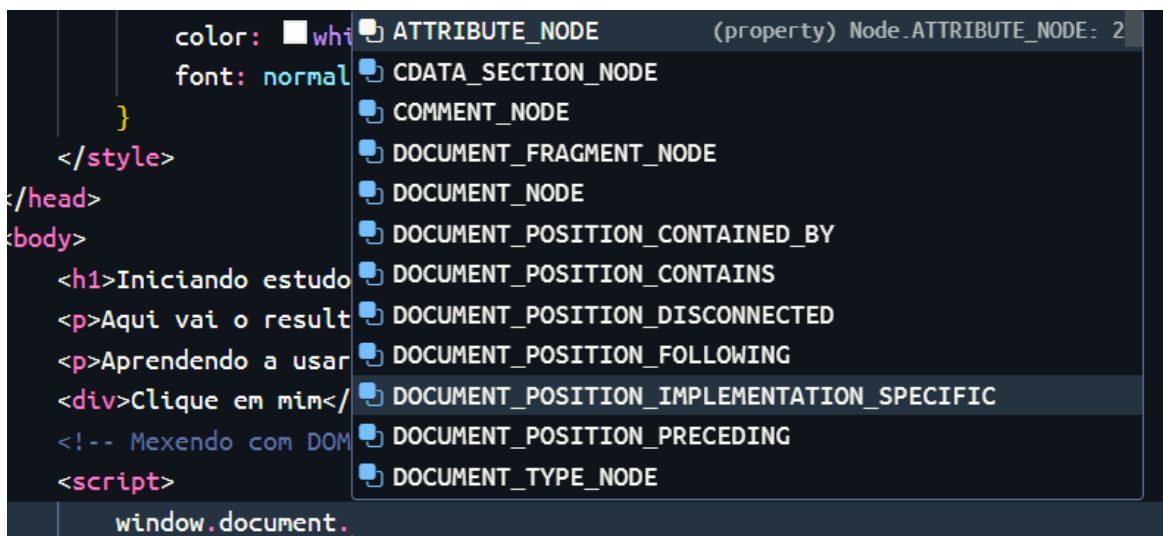
- window (janela);
 - location (diz a localização do seu site; qual é a URL, qual é página atual, qual foi a página anterior...)
 - document
 - html
 - head
 - meta
 - title
 - body

- h1
- p
 - strong
- div
- history (guarda de onde você veio e para onde você vai)

Elementos dentro de `window`:



Elementos dentro de `document`:

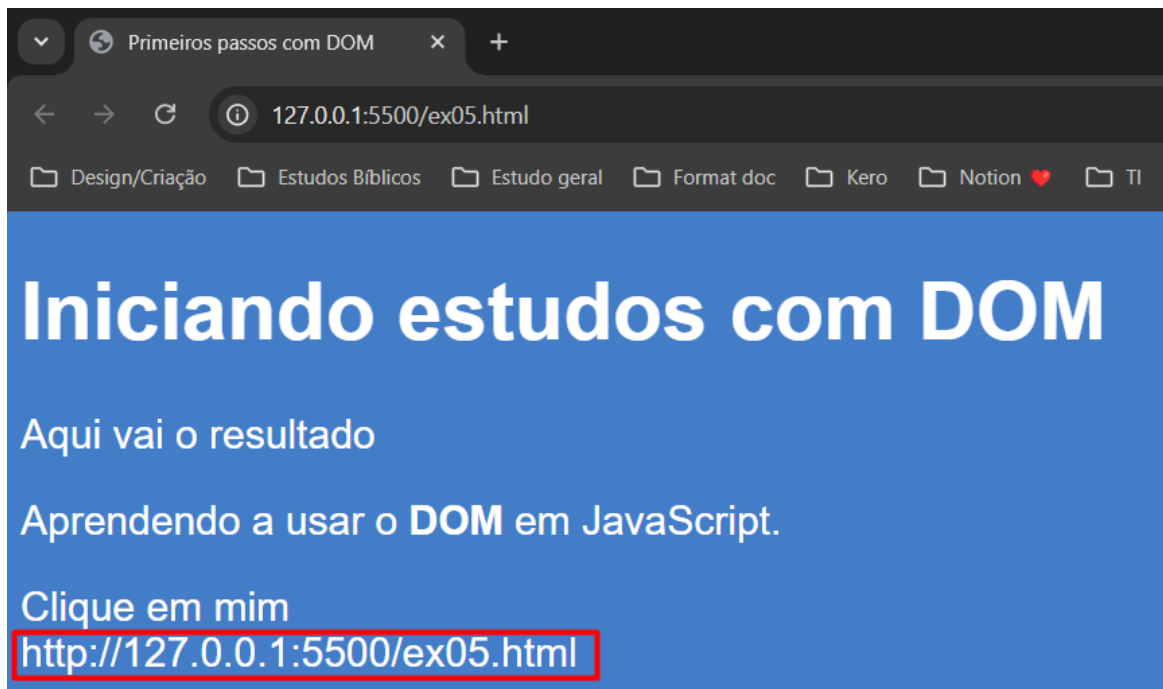


Ex.:

Código:

```
21 <script>
22   window.document.write(window.document.URL) //puxa a URL do documento
23 </script>
```

Resultado:



Técnicas de acesso aos elementos do JS:

- Por marca:

```
getElementsByTagName()
```

- Por ID:

```
getElementById()
```

- Por nome

```
getElementsByName()
```

- Por classe

```
getElementsbyClassName()
```

- Por seletor

```
querySelector()
```

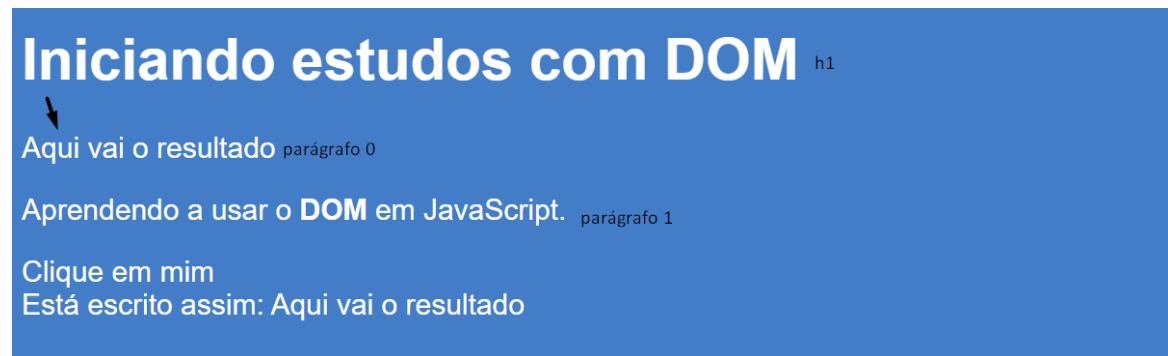
Selecionando elementos

Ex.:

Código:

```
24 // fazendo a seleção de elementos
25 var p1 = window.document.getElementsByTagName('p')[0] // seleciona parágrafos (no caso, vai selecionar o parágrafo zero)
26 window.document.write('Está escrito assim: ' + p1.innerText) // escreve na tela o texto do 1o parágrafo
```

Resultado:



Obs.:

- InnerHTML: pego o trecho de código HTML formatado.
- InnerText: pega o trecho de código HTML sem formatação.


Exemplos:

Com InnerHTML:

```
15 <body>
16   <h1>Iniciando estudos com DOM</h1>
17   <p>Aqui vai o resultado</p>
18   <p>Aprendendo a usar o <strong>DOM</strong> em JavaScript.</p> Parágrafo 1
19   <div>Clique em mim</div>
20   <!-- Mexendo com DOM -->
21   <script>
22     /* window.document.write(window.document.URL) */ //puxa a URL do documento
23
24     // fazendo a seleção de elementos
25     var p1 = window.document.getElementsByTagName('p')[1] // seleciona parágrafos (no caso, vai selecionar o parágrafo 1)
26     /* window.document.write('Está escrito assim: ' + p1.innerText) */ // escreve na tela o texto do 1o parágrafo
27
28     window.document.write(p1.innerHTML) // escreve o parágrafo 1 exatamente com a mesma formatação HTML
29   </script>
30 </body>
```

Iniciando estudos com DOM

Aqui vai o resultado

Aprendendo a usar o **DOM** em JavaScript.  Parágrafo 1

Clique em mim


Aprendendo a usar o **DOM** em JavaScript.

Com InnerText:

```
15 <body>
16   <h1>Iniciando estudos com DOM</h1>
17   <p>Aqui vai o resultado</p>
18   <p>Aprendendo a usar o <strong>DOM</strong> em JavaScript.</p>
19   <div>Clique em mim</div>
20   <!-- Mexendo com DOM -->
21   <script>
22     /* window.document.write(window.document.URL) */ //puxa a URL do documento
23
24     // fazendo a seleção de elementos
25     var p1 = window.document.getElementsByTagName('p')[1] // seleciona parágrafos (no caso, vai selecionar o parágrafo 1)
26     /* window.document.write('Está escrito assim: ' + p1.innerText) */ // escreve na tela o texto do 1o parágrafo
27
28     /* window.document.write(p1.innerHTML) */ // escreve o parágrafo 1 exatamente com a mesma formatação HTML
29     window.document.write(p1.innerText) // escreve o parágrafo 1 sem a formatação HTML
30   </script>
31 </body>
```

Iniciando estudos com DOM

Aqui vai o resultado

Aprendendo a usar o **DOM** em JavaScript.  Parágrafo 1

Clique em mim

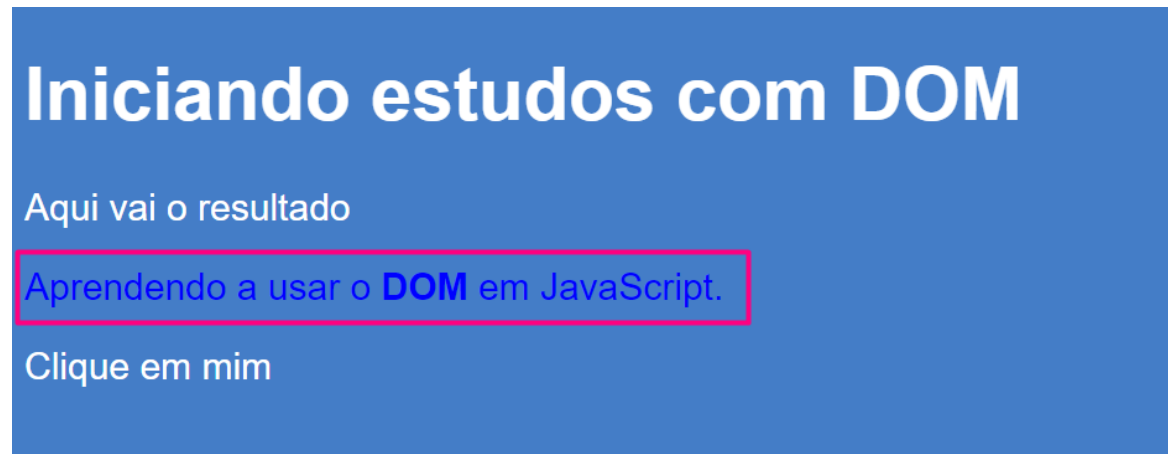
Aprendendo a usar o **DOM** em JavaScript.

Mudando a cor de um parágrafo com JS

Código:

```
31 // Mudando a cor de um parágrafo
32 p1.style.color = 'blue'
33 </script>
```

Resultado:

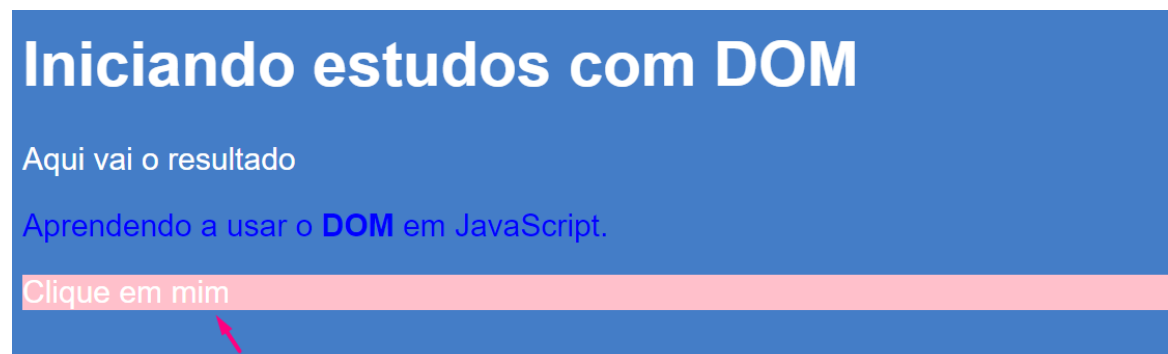


Mexendo com `getElementById` :

Código:

```
34 // Mexendo com getElementById
35 var d = window.document.getElementById('msg')
36 d.style.background = 'pink'
```

Resultado:



Mexendo com `getElementByName` :

```
<body>
  <div name="msg"> Texto </div>
  <script>
    var d = window.document.getElementsByName('msg')
```

```
[0]
</script>
</body>
```

Mexendo com `getElementByClassName` :

```
<body>
  <div class="msg"> Texto </div>
  <script>
    var d = window.document.getElementsByClassName
    ('msg')[0]
  </script>
</body>
```

Mexendo com seletores:

QuerySelector: serve para "economizar" linhas de comando

```
19      <div id="msg">Clique em mim</div>
```

```
40      // Usando querySelector
41      var d = window.document.querySelector('div#msg')
42      d.style.background = 'magenta'
```

Iniciando estudos com DOM

Aqui vai o resultado

Aprendendo a usar o **DOM** em JavaScript.

Clique em mim

LEMBRANDO QUE (Nada a ver com JS, mas sim com HTML):

- Id → .

- Class → #

▼ Aula 10) Eventos DOM

Pasta de exercícios relacionada: aula10

Com o DOM, acessamos os elementos da nossa página com JavaScript.

Evento é tudo o que pode acontecer com um elemento HTML.

Tipos de eventos:

- Mouse:
 - mouseenter
 - mousedown
 - click
 - mousemove
 - mouseup
 - mouseout

Para que um evento seja disparado, é necessário usar funções.

Função: conjunto de códigos/linhas que só serem executadas quando um evento ocorrer. Toda função funciona dentro de um bloco, que é representado por `{ }`. Ex. de bloco:

```
function {  
  
    // bloco função anônima  
  
}
```

No JS, função que não tem nome é chamada de anônima e é permitido. Mas para que o método possa funcionar, é necessário dar um título à função.

```
function ação(parâmetros){  
  
    // bloco função com nome
```



```
}
```

Os eventos podem ser disparados na parte do JS ou no HTML. Depende de como você preferir.

Ex. de disparos de eventos:

Código:

```
19 <body>
20   <!-- disparando um evento com um método dentro do elemento HTML-->
21   <div id="area" onclick="clicar()"> <!-- Quando eu clicar em cima da div... -->
22     <p>Interaja...</p>
23   </div>
24
25   <!-- JS -->
26   <script>
27     // Criando o bloco
28     function clicar() {
29       // Configurando o método clicar() para o que acontecerá quando a ação for executada
30       var area = window.document.getElementById('area')
31       area.innerText = 'Clicou!'
32       // O que acontece quando o usuário clicar na div: o texto da div "area" mudará para "Clicou!" quando for clicada.
33     }
34   </script>
35 </body>
```

Resultado:

Antes:



Depois:



Exemplo 2:

Código:

```
20 <!-- disparando eventos com métodos dentro do elemento HTML-->
21 <div id="area" onclick="clicar()" onmouseenter="entrar()" onmouseleave="sair()"> <!-- Quando eu executar essas ações... -->
22 <p>Interaja...</p>
23 </div>
24
25 <!-- JS -->
26 <script>
27     // Criando o bloco
28
29 > /* Modelo 1 - 1o exemplo...
37
38     /* Modelo 2 - 2o exemplo */
39     var area = window.document.getElementById('area') // Estando do "lado de fora" serve para todos os outros métodos
40     function clicar(){
41         area.innerText = 'Clicou!'
42     }
43
44     function entrar(){
45         area.innerText = 'Entrou!'
46     }
47
48     function sair(){
49         area.innerText = 'Saiu!'
50     }
51 </script>
52 </body>
```

Execução:

Ao passar o mouse em cima da div:



Ao retirar o mouse de cima da div:



Ao clicar na div:



Pra mudar a cor de fundo de um elemento HTML ao executar alguma ação:

```
// Dentro da função:  
nomeElemento.style.background = 'cor'
```

Disparando eventos pelo JS (os do exemplo acima foram pelo HTML):

```
22 <div id="area"> <!-- Quando eu executar essas ações... -->  
23 <p>Interaja...</p>  
24 </div>  
25  
26 <!-- JS -->  
27 <script>  
28 // Criando o bloco  
29  
30 > /* Modelo 1 - 1o exemplo...  
31  
32  
33  
34  
35  
36  
37  
38  
39 /* Modelo 2 - 2o exemplo */  
40 var area = window.document.getElementById('area') // Estando do "lado de fora" serve para todos os outros métodos  
41 /* Disparando eventos pelo próprio JS */  
42 area.addEventListener('click', clicar) // quando a div for clicada, o método clicar() será chamado  
43 area.addEventListener('mouseenter', entrar) // quando o mouse parar em cima da div, o método entrar() será chamado  
44 area.addEventListener('mouseout', sair) // quando o mouse sair de cima da div, o método sair() será chamado  
45  
46 function clicar(){  
47   area.innerText = 'Clicou!' // Muda o texto da div ao clicar nela  
48   area.style.background = 'red' // Muda o cor do fundo da div ao clicar nela  
49 }  
50  
51 function entrar(){  
52   area.innerText = 'Entrou!' // Muda o texto da div ao passar o mouse em cima dela  
53 }  
54  
55 function sair(){  
56   area.innerText = 'Saiu!' // Muda o texto da div ao tirar o mouse de cima dela  
57 }  
58 </script>
```

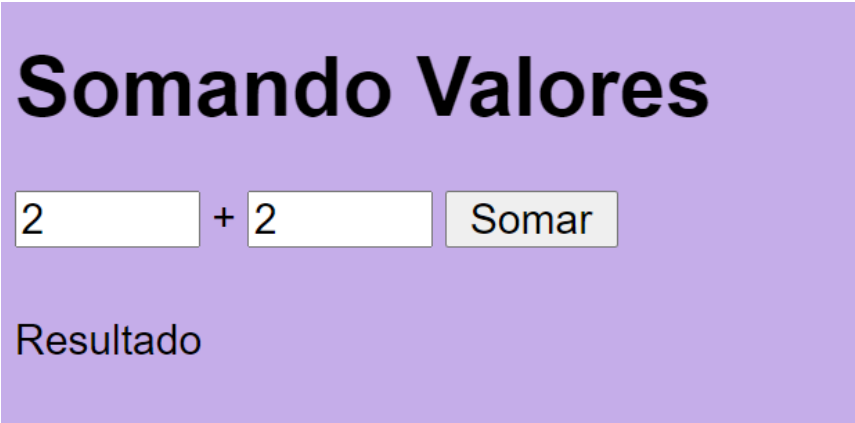
Obs.: a ferramenta "Inspeccionar" / DevTools do navegador aponta erros que o JS não tem costume de apontar.

Exercício 07 da aula 10 (Soma de valores)

Código:

```
23 <body>
24   <h1>Somando Valores</h1>
25   <input type="number" name="txtn1" id="txtn1"> +
26   <input type="number" name="txtn2" id="txtn2">
27   <input type="button" value="Somar" onclick="somar()"> <!-- Quando o botão "Somar" for clicado, a função somar() será chamada e as
    devidas alterações serão feitas de acordo a mesma -->
28   <div id="res">Resultado</div>
29
30   <!-- Mexendo com JS -->
31   <script>
32     // Criando a função para somar valores
33     function somar(){
34       var tn1 = window.document.getElementById('txtn1') // pega o valor do elemento txtn1
35       var tn2 = window.document.getElementById('txtn2') // pega o valor do elemento txtn2
36       var res = window.document.getElementById('res') // pega o conteúdo que está dentro da div 'res'
37       /* Outra maneira de usar:
38       var tn2 = window.document.querySelector('input#txtn2') // pega o valor do elemento txtn2
39       */
40
41       // Convertendo o valor dos txts de textos para números:
42       var n1 = Number(tn1.value)
43       var n2 = Number(tn2.value)
44
45       // Somando os valores:
46       s = n1 + n2
47
48       // Apresentando a soma na div 'res':
49       res.innerHTML = 'A soma entre ${n1} e ${n2} é igual a <strong>${s}</strong>.'
50     }
51   </script>
```

Resultado:



The screenshot shows a web application with a purple background. At the top, the title "Somando Valores" is displayed in a large, bold, black font. Below the title, there is a form consisting of two input fields, each containing the number "2", followed by a plus sign "+", another input field containing "2", and a button labeled "Somar". Below this form, the word "Resultado" is written in a large, bold, black font.

Somando Valores

+

A soma entre 2 e 2 é igual a **4**.