

Trilha de aprendizagem 02

Objetivo: Encontrar e substituir conteúdo de elementos html com javascript através da interface DOM.

Introdução

O elemento HTML é tudo, desde a **tag inicial** até a **tag final**:

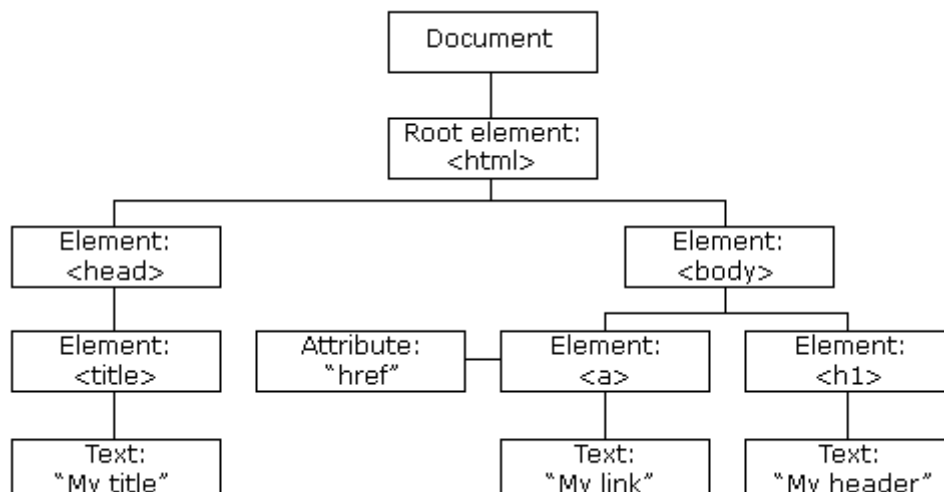
`< tagname >` O conteúdo vai aqui... `< /tagname >`

Os **seletores** são usados para "encontrar" (ou selecionar) os elementos HTML. Sendo assim, é possível selecionar elementos em HTML pelo ID, tag, classe e outros. A seleção de elementos HTML é realizada com objetivo de alterar seus conteúdos. Os objetos em HTML são selecionados e manipulados via DOM – Modelo de Objeto de Documento.

DOM - Modelo de Objeto de Documento

https://www.w3schools.com/js/js_htmlDOM.asp

Com o HTML DOM, o JavaScript pode acessar e alterar todos os elementos de um documento HTML. o HTML DOM é um padrão de como obter, alterar, adicionar ou excluir elementos HTML. No DOM, todos os elementos HTML são definidos como **objetos**. A seguir a árvore de objetos DOM:



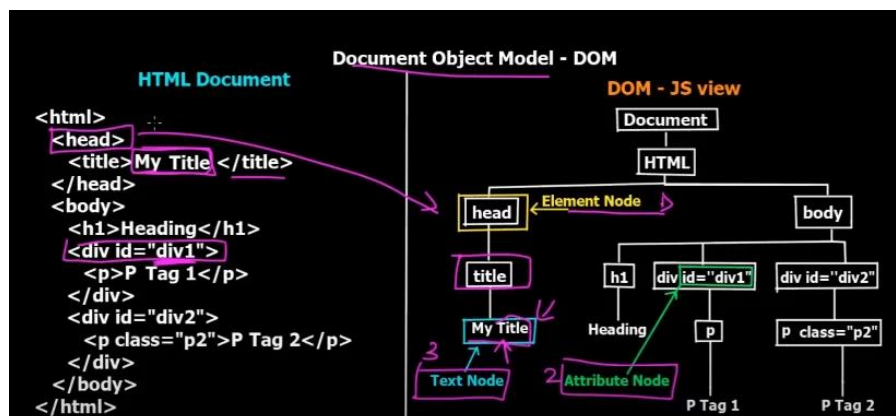
A interface de programação são as propriedades e métodos de cada objeto.

Uma **propriedade** é um valor que você pode obter ou definir (como alterar o conteúdo de um elemento HTML).

Um **método** é uma ação que você pode realizar (como adicionar ou excluir um elemento HTML).

Exemplo de **método**: getElementById

Exemplo de **propriedade**: InnerHTML.



Interatividade em Páginas Web

1 – Crie a página abaixo e observe o uso do método `getElementById` e da propriedade `innerHTML`.

```
<!DOCTYPE html>
<html>
  <body>
    <h2>My First Page</h2>
    <p id="demo"></p>
    <script>
      document.getElementById("demo").innerHTML = "Hello World!";
    </script>
  </body>
</html>
```

Encontrando elementos e substituindo conteúdo:

2 – Abra o editor de código JavaScript (VsCode). Abra seu projeto `javaScript`. Crie uma nova página HTML (`elementos.html`) e coloque a seguinte tag nela:

```
<p id="meuID"> Esse parágrafo será selecionado pelo ID. </p>
```

3 – Seguindo vamos manipular o conteúdo do elemento através de sua seleção por id. O método para obtenção do ID é o `getElementById`. Coloque o seguinte código `javaScript` logo abaixo do parágrafo:

```
<script>
  let elemento = document.getElementById("meuID");
  elemento.style.color = "red"; // Modifica a cor do texto para vermelho
</script>
```

4 – O próximo passo vai ser realizar a seleção por tag. Para selecionar elementos por tag, o método é `getElementsByTagName`. Adicione mais dois parágrafos no seu código.

```
<p>Este é o primeiro parágrafo.</p>
<p>Este é o segundo parágrafo. </p>
```

5 – Na sequência, insira o seguinte código em seu `javaScript` para manipular os parágrafos por tag:

```
let elementos = document.getElementsByTagName("p");
for (var i = 0; i < elementos.length; i++)
  elementos[i].style.color = "blue"; // Modifica a cor do texto para azul
```

6 – O próximo passo é a modificação do conteúdo por classe. O Método é o `getElementsByClassName`. Adicione o seguinte código em seu arquivo:

```
<p class="minhaClasse">Este é o primeiro parágrafo com a classe "minhaClasse".</p>
<p class="minhaClasse">Este é o segundo parágrafo com a classe "minhaClasse".</p>
```

7 – Seguindo, complemente seu `javaScript`:

```
let elementosc = document.getElementsByClassName("minhaClasse");
for (var i = 0; i < elementosc.length; i++)
  elementosc[i].style.color = "green"; // Modifica a cor do texto para verde
```

8 – Existem outras formas de selecionar elementos HTML: seleção de atributos, seleção de pseudo-classes, seleção de descendentes, seleção combinada e seleção universal. Para essas seleções são utilizados os métodos `javaScript`: `querySelector` ou `querySelectorAll`.

Exemplo por atributo:

```
// HTML
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

Interatividade em Páginas Web

```
// javascript
let elementoatr = document.querySelector("[href='https://www.w3schools.com']");
elementoatr.style.color = "purple"; // Modifica para a cor roxo o link
```

Exemplo por Pseudo classes:

```
// javascript
let elementopseudo = document.querySelector("p:first-child");
elementopseudo.style.color = "orange"; // Modifica a cor do texto para laranja
```

A pseudo classe first-child se refere ao primeiro parágrafo filho da árvore DOM. Nesse caso, o primeiro parágrafo da página vai ter sua cor alterada para laranja.

Exemplo por descendentes:

```
// HTML

<div> <p>Este parágrafo dentro da div será selecionado.</p> </div>

<p>Este parágrafo fora da div não será selecionado.</p>

// javascript

var elementodesc = document.querySelector("div p");
elementodesc.style.color = "purple"; // Modifica a cor do texto para marrom
```

Exemplo seleção combinada:

```
// HTML

<h1>Este é um cabeçalho</h1>
<p>Este parágrafo imediatamente após o cabeçalho será selecionado.</p>
<p>Este parágrafo não será selecionado.</p>

// javascript

let elementocomb = document.querySelector("h1 + p");
elementocomb.style.color = "pink"; // Modifica a cor do texto para rosa
```

Exemplo seleção Universal:

```
// javascript
var elementosall = document.querySelectorAll("*");
for (var i = 0; i < elementosall.length; i++)
    elementosall[i].style.border = "1px solid black"; // Adiciona uma borda preta a todos os elementos
```

Exemplos de modificação de atributo:

```
// HTML

<p id="dom" title="Objetos em HTML"> Aprenda DOM </p>

// javascript

document.getElementById("dom").title = "Objetos DOM";

// HTML



// javascript

document.getElementById("imgJava").src = "javascript.png";
```



Interatividade em Páginas Web

9 – Criando novos elementos. Para adicionar um novo elemento, primeiro cria-se ele, em seguida adiciona-se este a um dos nós existentes da árvore DOM. Faça o exemplo abaixo:

```
<div id="div1">
  <p id="p1">Este é o primeiro parágrafo. </p>
  <p id="p2">Este é o segundo parágrafo. </p>
</div>

<script>
  // criar o novo elemento:
  const paragrafo3 = document.createElement("p");

  // Para adicionar texto ao elemento <p>, primeiro deve ser criado o nó de texto.
  const noTexto = document.createTextNode("Terceiro parágrafo criado.");

  // Adicionar o no de texto ao parágrafo:
  Paragrafo3.appendChild(noTexto);

  // Adicionar o elemento <p> a um elemento existente.
  const element = document.getElementById("div1");
  element.appendChild(paragrafo3);
</script>
```

10 – Removendo elementos. Para remover um elemento, basta localiza-lo e em seguida usar o método *remove()*. Faça o exemplo abaixo:

```
<script>
  const elmnt = document.getElementById("p1");
  elmnt.remove();
</script>
```

11 – Substituindo elementos. Para substituir um elemento usa-se o método *replaceChild()*.

```
<script>
  const child = document.getElementById("p2");
  element.replaceChild(paragrafo3, child);
</script>
```

Interatividade em Páginas Web

VEJA COMO FICOU O CÓDIGO FINAL DA PÁGINA COM OS EXEMPLOS:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Manipulando elementos</title>
</head>
<body>
  <p id="meuID"> Esse parágrafo será selecionado pelo ID. </p>

  <p>Este é o primeiro parágrafo.</p>
  <p>Este é o segundo parágrafo.</p>

  <p class="minhaClasse">Este é o primeiro parágrafo com a classe "minhaClasse".</p>
  <p class="minhaClasse">Este é o segundo parágrafo com a classe "minhaClasse".</p>

  <a href="https://www.w3schools.com">Visit W3Schools</a>

  <div> <p>Este parágrafo dentro da div será selecionado.</p> </div>

  <p>Este parágrafo fora da div não será selecionado.</p>

  <h1>Este é um cabeçalho</h1>
  <p>Este parágrafo imediatamente após o cabeçalho será selecionado.</p>
  <p>Este parágrafo não será selecionado.</p>

  <p id="dom" title="Objetos em HTML"> Aprenda DOM <p>

  <div id="div1">
    <p id="p1">Este é o primeiro parágrafo. </p>
    <p id="p2">Este é o segundo parágrafo. </p>
  </div>

  <script>

    document.getElementById("dom").title = "Objetos DOM";

    document.getElementById("imgJava").src = "javascript.png";

    let elemento = document.getElementById("meuID");
    elemento.style.color = "red"; // Modifica a cor do texto para vermelho

    let elementos = document.getElementsByTagName("p");
    for (var i = 0; i < elementos.length; i++)
      elementos[i].style.color = "blue"; // Modifica a cor do texto para azul

    let elementosc = document.getElementsByClassName("minhaClasse");
    for (var i = 0; i < elementosc.length; i++)
      elementosc[i].style.color = "green"; // Modifica a cor do texto para verde

    let
      elementoatr =
document.querySelector("[href='https://www.w3schools.com']");
    elementoatr.style.color = "purple"; // Modifica para a cor roxo o link

    let elementopseudo = document.querySelector("p:first-child");
    elementopseudo.style.color = "orange"; // Modifica a cor do texto para
laranja

    let elementodesc = document.querySelector("div p");
    elementodesc.style.color = "purple"; // Modifica a cor do texto para roxo

    let elementocomb = document.querySelector("h1 + p");
    elementocomb.style.color = "pink"; // Modifica a cor do texto para rosa

    var elementosall = document.querySelectorAll("*");
    for (var i = 0; i < elementosall.length; i++)
      elementosall[i].style.border = "1px solid black"; // Adiciona uma borda
preta a todos os elementos
```



Interatividade em Páginas Web

```
// criar o novo elemento:
const paragrafo3 = document.createElement("p");

// Para adicionar texto ao elemento <p>, primeiro deve ser criado o nó
de texto.

const noTexto = document.createTextNode("Parágrafo criado.");

// Adicionar o nó de texto ao parágrafo:
paragrafo3.appendChild(noTexto);

// Adicionar o elemento <p> a um elemento existente.
const element = document.getElementById("div1");
element.appendChild(paragrafo3);

const elmnt = document.getElementById("p1");
elmnt.remove();

const child = document.getElementById("p2");
element.replaceChild(child, paragrafo3);

</script>

</body>
</html>
```