

INSTITUTO FEDERAL

Paraná

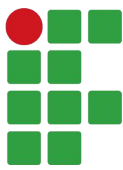
Campus Paranaguá

JavaScript

DOM - Document Object Model

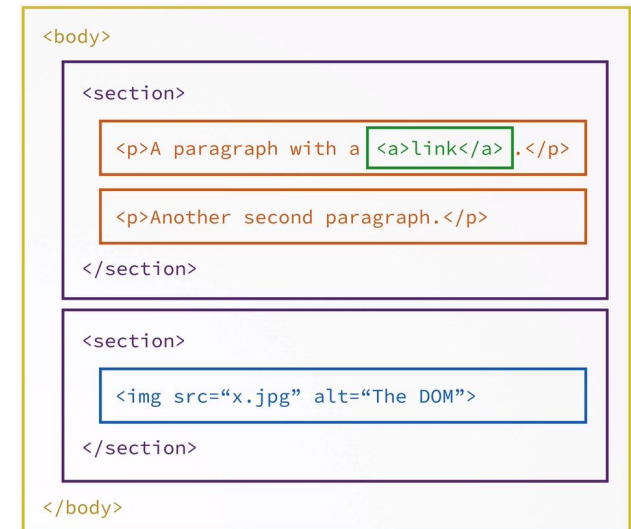
Desenvolvimento Web

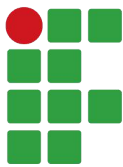
Prof. Diego Stiehl



DOM

- **Document Object Model**
 - Representação estruturada de documentos HTML
- É utilizado para conectar o HTML com scripts
 - O JavaScript consegue ler e alterar o DOM
- Para cada “caixa” do HTML há um objeto no DOM que podemos acessar e interagir





document

- No JavaScript vinculado a uma página HTML, o objeto `document` representa a raiz do DOM
 - Todo acesso ao DOM parte dele
 - Todos elementos pertencem a ele
 - Para o JS, document é a **página HTML**
- Enxergamos o documento como um objeto JS

getElementById()

- Buscar um elemento na página
 - Usa o atributo id como base
 - Retorna um objeto que representa o elemento
- Exemplos:

```
let elemento1 = document.getElementById('id-elemento');  
let elemento2 = document.getElementById('id-outro-elemento');
```

```
<p id="id-elemento">Texto</p>
```

```
<div id="id-outro-elemento">Uma div</div>
```

querySelector()

- Método para selecionar subelementos
- Podemos procurar por qualquer elemento
 - Utiliza a sintaxe do CSS

- Exemplos:

```
let elemento1 = document.querySelector('#id-elemento');  
let elemento2 = document.querySelector('.classe');
```

```
<p id="id-elemento">Texto</p>  
<div class="classe">Uma div</div>
```

- Seleciona apenas primeiro elemento do tipo
 - Para array, usar `document.querySelectorAll('div')`

querySelectorAll

- Retorna um vetor com todos os elementos que se enquadram com o seletor

```
const itens = document.querySelectorAll('li');
```

```
const primeiroItem = itens[0];
```

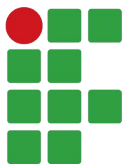
```
const segundoItem = itens[1];
```

```
const terceiroItem = itens[2];
```

```
console.log(primeiroItem);
```

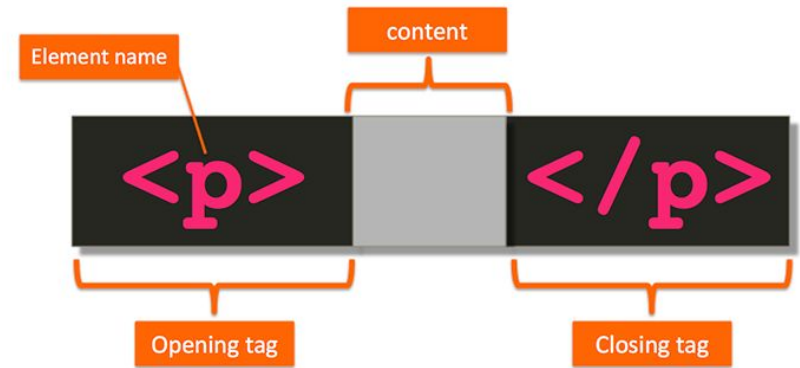
```
console.log(segundoItem);
```

```
console.log(terceiroItem);
```

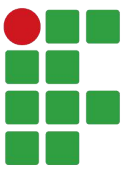


textContent

- Permite ler e escrever no conteúdo de um elemento HTML selecionado
 - Tudo que está entre `<>` e `</>`
 - **Somente texto puro**

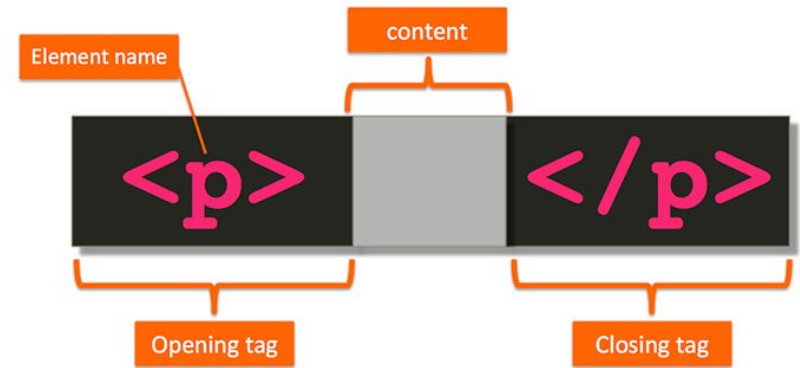


```
let elemento = document.querySelector('.container');  
console.log(elemento.textContent); // Lendo o conteúdo  
elemento.textContent = 'Alterado'; // Alterando o conteúdo
```



innerHTML

- Permite ler e escrever no conteúdo de um elemento HTML selecionado
 - Tudo que está entre `<>` e `</>`
 - **Permite inserir novas tags HTML**



```
let elemento = document.querySelector('.container');  
console.log(elemento.innerHTML); // Lendo o conteúdo  
elemento.innerHTML = '<em>Itálico</em>'; // Alterando
```


Alterando Estilos

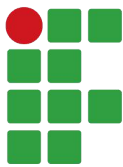
- Podemos alterar estilos CSS através do DOM
- Atributo `style`
 - Irá aplicar estilos inline via JavaScript
 - Usar mesmo nome das **propriedades do CSS**
 - Mas com **camelCase**

```
elemento.style.display = 'none';  
elemento.style.backgroundColor = '#ff5731';  
elemento.style.textAlign = 'center';
```

Atribuição de Classes

- Podemos manipular as classes (atributo class) de qualquer elemento HTML
 - Adicionar ou remover classes em um elemento
 - Trabalho em conjunto com as regras do CSS
- Utilizar propriedade `classList`
 - Métodos `add()`, `remove()` e `toggle()`

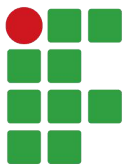
```
elemento.classList.add('principal');  
elemento.classList.remove('principal');  
elemento.classList.toggle('ativo'); // add ou remove
```



Inputs

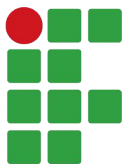
- Para ler ou alterar o valor de um input, utilizamos a propriedade `value`

```
let cidade = document.querySelector('.campo-cidade');  
console.log(cidade.value); // Lê o conteúdo  
cidade.value = "Paranaguá"; // Altera o conteúdo
```



Eventos

- Quando algo acontece no HTML, temos um evento (event)
 - Um clique em um botão
 - Submissão de formulário
 - Usuário aperta uma tecla
 - ...
- Podemos “escutar” por eventos no DOM
 - Registramos **Event Listeners** para **Events** específicos em elementos pré definidos



Eventos

- Para registrar um Listener, dizemos qual o evento “escutado” e que função deve ser executada
 - Pode ser uma função predefinida ou anônima

```
function duploClique() {  
    console.log('Duplo clique!');  
}
```

```
let elemento = document.querySelector('#botao');  
elemento.addEventListener('dblclick', duploClique);
```

```
elemento.addEventListener('click', function() {  
    console.log('Clicou!');  
});
```

Mais Eventos

- Para conhecer todos eventos, consulte:
 - [Event reference | MDN](#)

Outras Propriedades

`input.checked`

`input.selectedIndex`

`imagem.src`

`link.href`

//... Tudo que se vê no HTML ...