



**INSTITUTO FEDERAL**

São Paulo

Câmpus São Carlos

## Aula 05

# Manipulação HTML/DOM

Prof. Tiago Henrique Trojahn

`tiagotrojahn@ifsp.edu.br`

# Manipulação de elementos HTML

```
var elemento = document.getElementsByTagName("div")[0];  
elemento.innerHTML = "Novo conteúdo do elemento";
```

.innerHTML: Modifica o conteúdo interno do elemento selecionado

# Manipulação de elementos HTML

```
var elemento = document.getElementsByTagName("div")[0];  
elemento.setAttribute("id", "ifsp");
```

.setAttribute(atributo, valor): Modifica (ou adiciona) o atributo para o elemento selecionado.

# Manipulação de elementos HTML

```
var elemento = document.getElementsByTagName("div")[0];  
elemento.getAttribute("id");
```

.getAttribute(atributo): Retorna o valor do atributo selecionado

Caso não exista: NULL

# Manipulação de elementos CSS

```
var elemento = document.getElementsByTagName("p")[0];
// Muda o background-color para vermelho
elemento.style.backgroundColor = "red";
// Muda a cor da fonte para branca
elemento.style.color = "white";
// Aumenta o tamanho da fonte em 5px
// Primeiro, usando o getComputedStyle recupera o valor CSS correto do elemento
// Segundo, usa o parseInt para converter '16px' para '16'
// Por fim, adiciona 20 a esse número e concatena com a string 'px'.
elemento.style.fontSize = parseInt(getComputedStyle(elemento)["font-size"]) + 20 + "px";
// Imprime no console o valor do 'background-color' do elemento.
console.log(getComputedStyle(elemento)["background-color"]);
```

# Manipulação de elementos HTML

```
// Cria um novo elemento HTML (p)
```

```
var novoParagrafo = document.createElement("p");
```

```
// Cria um elemento de texto
```

```
var conteudo = document.createTextNode("Novo parágrafo");
```

```
// Adiciona um elemento HTML dentro de outro elemento
```

```
novoParagrafo.appendChild(conteudo);
```

```
// Remove um filho de um elemento
```

```
novoParagrafo.removeChild(conteudo);
```

# Exercício

- Dada um array de elementos, crie a função que adicione todos os elementos do array como `<li>` dentro de uma listagem não-ordenada no website...
- Considere que a `<ul>` NÃO EXISTE (ainda...)

`<ul>`

`<li>Pêssego</li>`

`<li>Uva</li>`

`<li>Maça</li>`

`<li>Limão</li>`

- Pêssego
- Uva
- Maça
- Limão

`</ul>`

# Exercício - Resolução

```
var itens = ["Pera", "Abacaxi", "Melância", "Abacate", "Jabuticaba"];  
var ul = document.createElement("ul");  
for (var i = 0; i < itens.length; i++) {  
    var conteudo = document.createTextNode(itens[i]);  
    var li = document.createElement("li");  
    li.appendChild(conteudo);  
    ul.appendChild(li);  
}  
// Ou simplesmente 'document.body.appendChild(ul)';  
document.getElementsByTagName("body")[0].appendChild(ul);
```



# Exercício de Fixação

```
<button>Adicionar</button>
<button>Remover</button>
<table border="1">
  <thead>
    <tr>
      <th>Coluna 1</th>
      <th>Coluna 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Linha 1</td>
      <td>Linha 1</td>
    </tr>
    <tr>
      <td>Linha 2</td>
      <td>Linha 2</td>
    </tr>
  </tbody>
</table>
```

- Considerando o código ao lado, associe o evento de *click* aos botões de Adicionar e Remover.
- Ao clicar em Adicionar, uma nova linha deve ser inserida no tbody (Linhas 3, Linha 4, etc...). O sistema **não deve permitir mais que quinze linhas no tbody**.
- Ao clicar em Remover, a última linha do tbody é removida. **Não deve haver menos que duas linhas no tbody**.