

**INSTITUTO FEDERAL**  
Sul-rio-grandense

Câmpus  
Charqueadas

# Programação Web I

Javascript e DOM

Prof. Pablo Werlang  
[pablowerlang@ifsul.edu.br](mailto:pablowerlang@ifsul.edu.br)

# Javascript e DOM

## Manipulação de elementos

- DOM - Document Object Model
  - Representação dos objetos que compõe o documento.
- Conceito de separação do Código
  - HTML, CSS e JS em arquivos separados

### HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Minha Página</title>
  <link rel="stylesheet" href="style.css">
  <script type='text/javascript' src='script.js'
async></script>
</head>
<body>
</body>
</html>
```

### CSS

```
body {
  color: red;
}
```

### JS

```
document.body.innerHTML = 'Hello World!';
```

# Javascript e DOM

## Manipulação de elementos

- Procurando elementos com querySelector

### HTML

```
<body>
  <h1></h1>
  <h2 id="subtitle"></h2>
  <div id="row">
    <span class="inside"></span>
    <span class="inside"></span>
    <span class="inside"></span>
  </div>
</body>
```

### CSS

Bônus: [Tutorial flexbox](#)

```
#row {
  display: flex;
}

#row .inside {
  border: 1px solid black;
  background-color: #00000030;
  padding: 5px 15px;
  margin: 0 10px;
}
```

. = class  
JS

# = ID

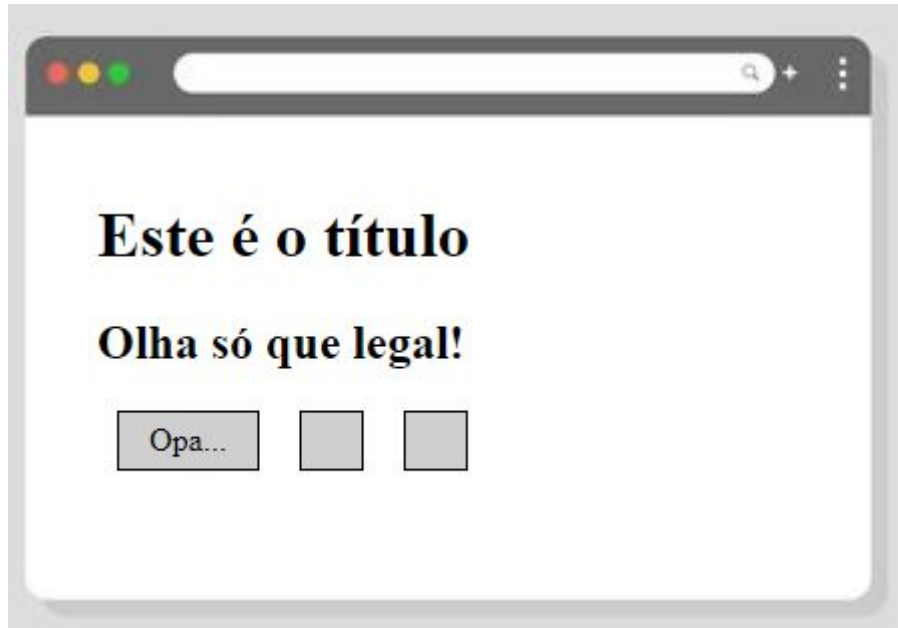
tag

```
document.querySelector('h1').innerHTML = 'Este é o título';
document.querySelector('#subtitle').innerHTML = 'Olha só que legal!';
document.querySelector('.inside').innerHTML = 'Opa...'; // vamos ver isso mais adiante
```

# Javascript e DOM

Manipulação de elementos

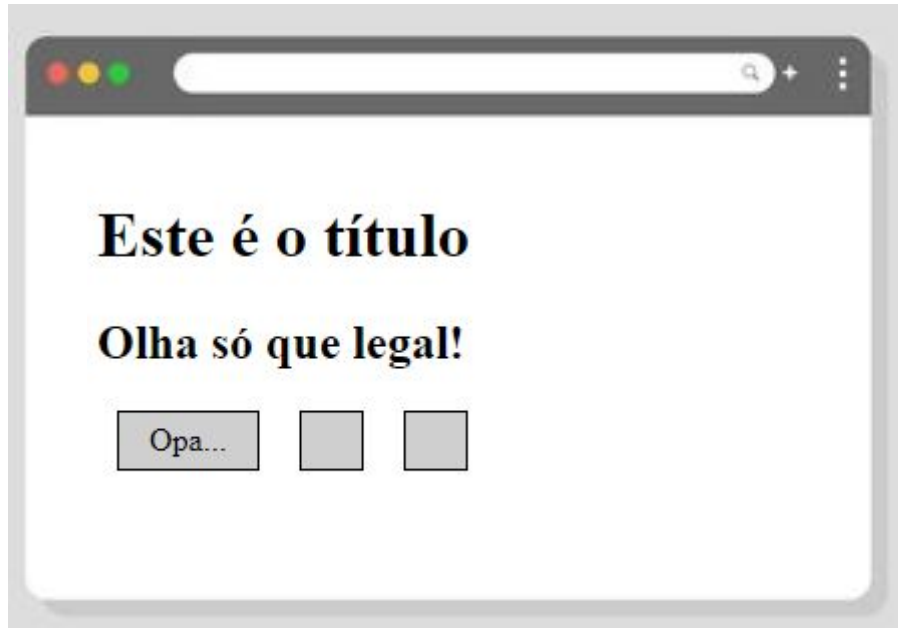
- Como resolver esse problema?



# Javascript e DOM

## Manipulação de elementos

- Como resolver esse problema?



**Veremos mais de uma maneira no futuro.**

**Por enquanto, nos atemos ao simples.**

# Javascript e DOM

Manipulação de elementos



HMTL

```
<div id="row">
  <span class="inside one"></span>
  <span class="inside two"></span>
  <span class="inside three"></span>
</div>
```

Não tem espaço

JS

```
document.querySelector('.inside.one').innerHTML = 'Primeiro';
document.querySelector('.inside.two').innerHTML = 'Segundo';
document.querySelector('.inside.three').innerHTML = 'Terceiro';
```

# Javascript e DOM

## Manipulação de elementos

```
#box {  
  width: 300px;  
  border: 1px solid black;  
  background-color: #323232;  
  padding: 15px 30px;  
  display: flex;  
  flex-direction: column;  
}  
  
#box #title {  
  display: flex;  
  margin: 5px 1px 20px 0px;  
  border: 1px solid white;  
  background-color: #a0a0a0;  
  color: black;  
  padding: 5px 15px;  
  justify-content: center;  
}  
  
#box #content {  
  color: white;  
  margin-bottom: 20px;  
}
```

```
<body>  
  <h1 id="title">Título</h1>  
  <div id="box">  
    <span id="title">Isso não era pra estar aqui</span>  
    <span id="content">Conteúdo da caixa</span>  
  </div>  
</body>
```

HMTL

JS

```
document.querySelector('#title').innerHTML = 'Título da caixa';
```

OPA!!!

O que deu errado?

# Javascript e DOM

## Manipulação de elementos

```
#box {  
  width: 300px;  
  border: 1px solid black;  
  background-color: #323232;  
  padding: 15px 30px;  
  display: flex;  
  flex-direction: column;  
}
```

```
#box #title {  
  display: flex;  
  margin: 5px 1px 20px 0px;  
  border: 1px solid white;  
  background-color: #a0a0a0;  
  color: black;  
  padding: 5px 15px;  
  justify-content: center;  
}
```

```
#box #content {  
  color: white;  
  margin-bottom: 20px;  
}
```

```
<body>  
  <h1 id="title">Título</h1>  
  <div id="box">  
    <span id="title">Isso não era pra estar aqui</span>  
    <span id="content">Conteúdo da caixa</span>  
  </div>  
</body>
```

HMTL

JS

```
document.querySelector('#box #title').innerHTML = 'Título da caixa';
```

Fazemos igual no CSS, colocamos  
ESPAÇO para indicar que queremos  
selecionar o "filho".



# Javascript e DOM

## Manipulação de elementos

### HTML

```
<body>
  <h1 id="title">Título</h1>
  <div id="box">
    <span id="title">CAIXA</span>
    <span id="content"></span>
  </div>
</body>
```

### JS

```
document.querySelector('#box #content').innerHTML =
'<ul><li>Primeiro item</li><li>Segundo item</li><li>Terceiro
item</li></ul>';
```

**Podemos também inserir HTML dentro de um elemento**



# Javascript e DOM

## Manipulação de elementos

- Podemos usar `" "`, `' '`, ou `` `` quando usando string no JS.
- A vantagem das aspas invertidas é que podemos colocar expressões dentro da string usando o `${}`.

### HTML

```
<body>
  <h1 id="title">Título</h1>
  <div id="box">
    <span id="title">CAIXA</span>
    <span id="content"></span>
  </div>
</body>
```

### JS

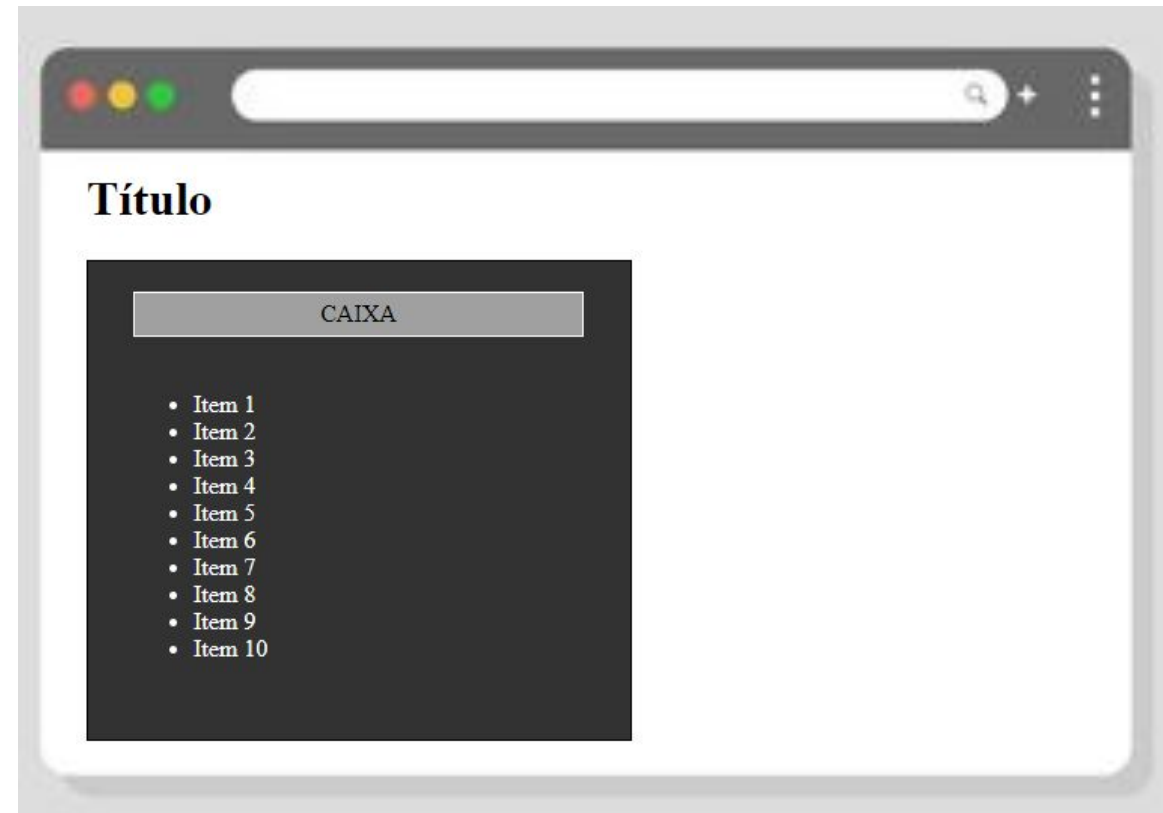
```
let value = prompt('Informe seu nome:');
document.querySelector('#box #content').innerHTML = `Bem-vindo ${value}!`;
```

# Javascript e DOM

## Manipulação de elementos

### Exercício

- Vamos criar uma caixa como na imagem ao lado.
- Pergunte para o usuário com um prompt quantos itens ele quer imprimir.
- Usando uma estrutura de repetição, escreva em uma variável o HTML de cada item da lista.
- Mostra no documento, dentro da caixa.



# Javascript e DOM

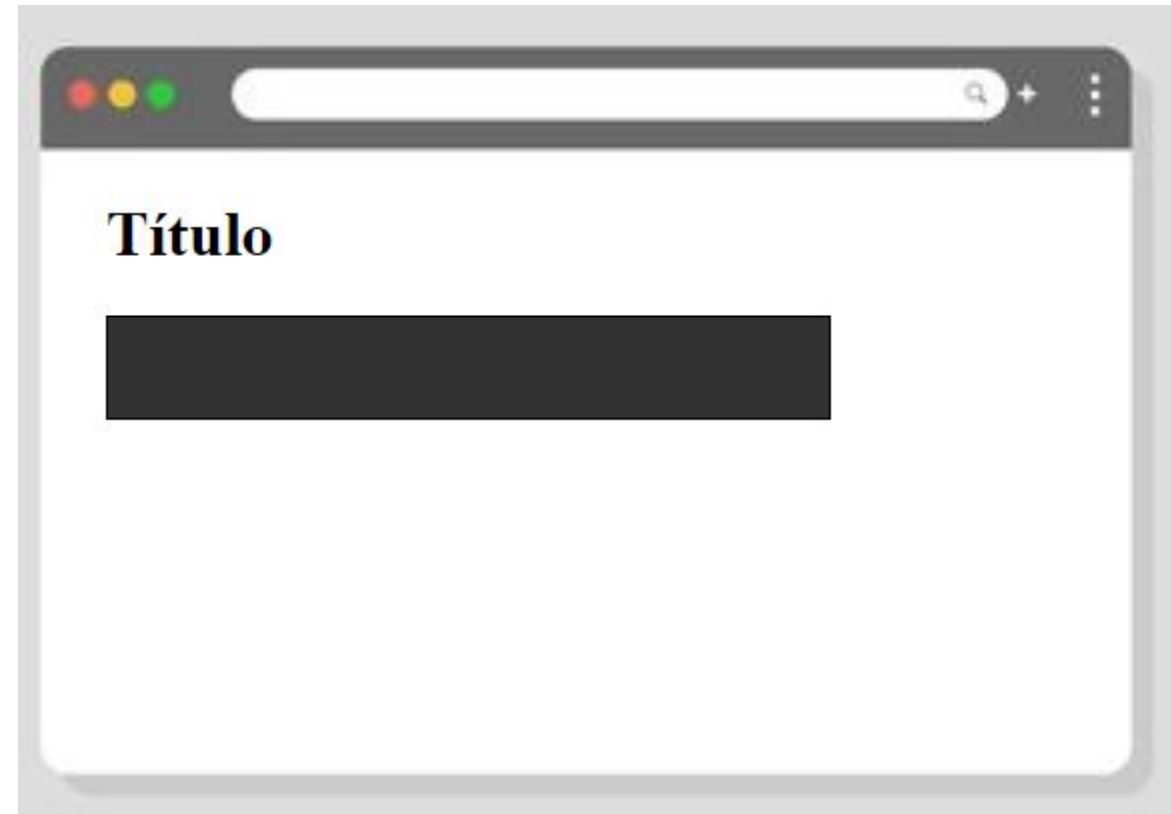
## Manipulação de elementos

### Excluindo elementos:

- Selecione usando o *querySelector*.
- Use a função *remove()*.

JS

```
document.querySelector('#box #title').remove();
```



# Javascript e DOM

## Manipulação de elementos

### Inserindo elementos:

- Existem várias funções para isso, mas podemos usar *insertAdjacentHTML*, que é bem simples e poderosa.

JS

```
const element = document.querySelector('#box #title');

element.insertAdjacentHTML('beforebegin', '<span  
class="red">Antes do elemento</span>');

element.insertAdjacentHTML('afterend', '<span  
class="red">Depois do elemento</span>');

element.insertAdjacentHTML('afterbegin', '<span  
class="red">No início do elemento</span>');

element.insertAdjacentHTML('beforeend', '<span  
class="red">No fim do elemento</span>');
```

```
.red {  
  color: red;  
}
```



# Javascript e DOM

## Manipulação de elementos

### Exercício:

- Usando somente JS (sem editar o arquivo HTML).
- Mostre na tela 10 caixas cinzas.
- Dentro das caixas os múltiplos de 5.
- Use uma repetição e o `insertAdjacentHTML` para inserir as caixas.
- Um emoji de alien 🛸 no início e fim da sequência.
- Siga o exemplo ao lado.



# Javascript e DOM

## Manipulação de elementos

### Outras funções de manipulação de DOM importantes:

- insertAdjacentElement - Parecido com insertAdjacentHTML, mas insere um objeto (elemento) ao invés de um texto.
- createElement - Cria um novo elemento que pode ser inserido no DOM com insertAdjacentElement.

```
const myDiv = document.createElement('div');  
myDiv.innerHTML = `

# 

document.body.insertAdjacentElement('beforeend', myDiv);
```

# Javascript e DOM

## Manipulação de elementos

### Outras funções de manipulação de DOM importantes:

- closest - busca o elemento "pai" mais próximo da string de busca (semelhante ao querySelector).

```
<div id="a">
  <div id="b">
    <div id="c"></div>
    <div id="d"></div>
  </div>
  <div id="b">
    <div id="e"></div>
    <div id="f"></div>
  </div>
</div>
```

```
let b = document.querySelector('#b');
let c = b.querySelector('#c');
```

```
let e = document.querySelector('#e');
let b = e.closest('#b');
```

Mais serão vistas na parte 2!



# Javascript e DOM

Soon!

Voltaremos com manipulação do DOM.

Ainda veremos:

- Como acessar, modificar, adicionar e excluir atributos dos elementos.
- Alterar o css usando JS.
- `querySelectorAll` para acessar todos elementos buscados.

**Aguardem a parte 2!**