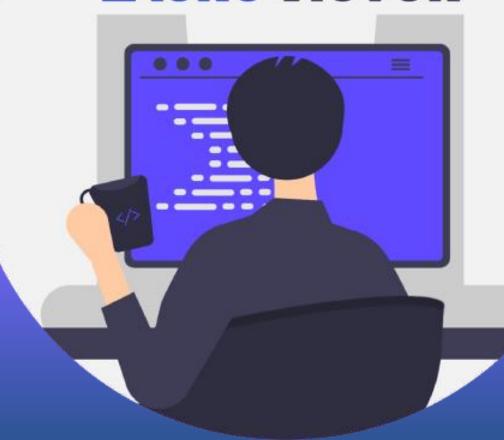
# LAURO VICTOR





#### Alteração de atributos

- Os atributos são valores que contêm informações adicionais sobre elementos do HTML. Normalmente, eles vêm em pares de nome/valor e podem ser essenciais dependendo do elemento.
- Alguns dos atributos do HTML mais comuns são os atributos src de uma tag img, o href de uma tag a, a tag class, id e style.



#### Alteração de atributos

- No JavaScript, temos quatro métodos para modificar atributos do elemento:
- hasAttribute()
- getAttribute()
- setAttribute()
- removeAttribute()



Alteração de atributos - hasAttribute()

• Devolve valores lógicos booleanos do tipo true ou false

var existe = element.hasAttribute('href');



Alteração de atributos - getAttribute()

Retorna o valor de um atributo especificado ou null caso não exista o atributo

var valorAtributo = element.getAttribute('href');



Alteração de atributos - setAttribute()

Adiciona ou atualiza o valor de um atributo especificado

element.setAttribute('href', 'index.html')



Alteração de atributos - removeAttribute()

Remove um atributo de um elemento

element.removeAttribute('href')



#### Alteração de atributos

 Finalmente, podemos modificar o atributo diretamente atribuindo um novo valor ao atributo como uma propriedade do elemento

```
element.href = "https://www.wikipedia.org/";
```



#### Alteração de atributos

- Os métodos hasAttribute() e getAttribute() são normalmente usados com declarações condicionais
- Os métodos setAttribute() e removeAttribute() são usados para modificar diretamente o DOM



#### Alteração de estilos

- A propriedade estilo representa os estilos embutidos em um elemento HTML. Muitas vezes, os estilos serão aplicados aos elementos através de uma folha de estilos. Porém, às vezes precisamos adicionar ou editar diretamente um estilo embutido.
- Uma opção para editar os estilos é com o setAttribute().

```
const div = document.querySelector('div');
div.setAttribute('style', 'text-align: center');
```



#### Alteração de estilos

No entanto, isso removerá todos os estilos embutidos do elemento.
 Uma vez que este provavelmente não é o efeito desejado, é melhor usar o atributo style diretamente

```
div.style.height = '100px';
div.style.width = '100px';
div.style.border = '2px solid black';
```



#### Alteração de estilos

- As propriedades CSS são escritas em kebab-case, que são palavras minúsculas separadas por traços. É importante notar que as propriedades CSS em kebab-case não podem ser usadas na propriedade de estilo do JavaScript.
- Em vez disso, elas serão substituídas por seu equivalente em camelCase, que é quando a primeira palavra é minúscula e todas as palavras subsequentes são maiúsculas. Em outras palavras, em vez de text-align, vamos usar textAlign para a propriedade de estilo do JavaScript.



#### Alteração de estilos

Kebab-case se tornando camelCase:

```
div.style.borderRadius = '50%';
div.style.display = 'flex';
div.style.justifyContent = 'center';
div.style.alignItems = 'center';
```



#### Alteração de classes

- O atributo classe corresponde aos seletores de classe CSS
- As classes CSS são usadas para aplicar estilos a vários elementos, ao contrário das IDs que só podem existir uma vez por página. No JavaScript, temos as propriedades de className e de classList para trabalhar com o atributo de classe



#### Alteração de classes - classname

Obtém ou define o valor de classe

var classe = element.className;



#### Alteração de classes - classname

 A propriedade className foi introduzida para evitar conflitos com a palavra-chave class encontrada no JavaScript e outras linguagens que têm acesso ao DOM. É possível usar o className para atribuir um valor diretamente à classe.

```
const div = document.querySelector('div');
div.className = "warning";
```



#### Alteração de classes - classname

Note que se alguma classe já existir no elemento, isso irá substituí-las.
É possível adicionar várias classes com espaço delimitado usando a
propriedade className, ou usá-las sem operadores de atribuição para
obter o valor da classe no elemento

```
const div = document.querySelector('div');
div.className = "warning";
```



#### Alteração de classes - classList

- Existem alguns métodos de manipulação da nossa lista de classes:
- Add
- Toggle
- Contains
- Replace
- Remove



#### Alteração de classes - Add

Adiciona um ou mais valores de classe

```
element.classList.add('active');
element.classList.add("first", "second", "third");
```



#### Alteração de classes - Toggle

 Adiciona a classe caso o elemento n\u00e3o a possua ou retira a classe caso o elemento j\u00e1 a possua

element.classList.toggle('active');



#### Alteração de classes - Contains

Verifica se o valor de classe existe

element.classList.contains('active');



#### Alteração de classes - Replace

 Substitui um valor de classe existente com um novo valor de classe element.classList.replace('old', 'new');



#### Alteração de classes - Remove

Remove um valor de classe

element.classList.remove('active');



# **Obrigado!**



