

# Para saber mais: trabalhando com atributos



25%

ATIVIDADES 3 DE 9

> DISCORD ALURA

FÓRUM DO CURSO

VOLTAR PARA DASHBOARD

> MODO CLARO



Agora, vamos falar sobre a manipulação de atributos de elementos no DOM usando JavaScript. Existem várias formas de manipular esses elementos, mas vamos nos concentrar em quatro métodos muito úteis e comuns: getAttribute, setAttribute, hasAttribute e removeAttribute.

## getAttribute:

O método getAttribute é usado para obter o valor de um atributo específico de um elemento HTML. Ele recebe um argumento, que é o nome do atributo que desejamos recuperar o valor. Por exemplo:

```
// HTML: <div id="meuElemento" data-info="Exemplo de a
const elemento = document.getElementById('meuElemento
const valorDoAtributo = elemento.getAttribute('data-in
console.log(valorDoAtributo); // Saída: "Exemplo de at</pre>
```

240.3k xp





25%

ATIVIDADES 3 DE 9

> DISCORD ALURA

FÓRUM DO CURSO

VOLTAR PARA DASHBOARD



Neste exemplo, o método getAttribute foi usado para obter o valor do atributo data-info do elemento com o ID "meuElemento".

#### setAttribute:

O método setAttribute é usado para definir ou modificar o valor de um atributo em um elemento HTML. Ele aceita dois argumentos: o primeiro é o nome do atributo que queremos definir ou modificar, e o segundo é o valor que queremos atribuir a esse atributo. Se o atributo já existir, o método setAttribute irá sobrescrevê-lo; caso contrário, ele criará um novo atributo. Por exemplo:

```
// HTML: Texto inicial
```

const paragrafo = document.getElementById('meuParagraparagrafo.setAttribute('id', 'paragrafoModificado');
paragrafo.setAttribute('data-novo-atributo', 'Novo vai')



240.3k xp





25%

ATIVIDADES 3 DE 9

> DISCORD ALURA

FÓRUM DO CURSO

VOLTAR PARA DASHBOARD



Após a execução deste código, o parágrafo terá seu ID alterado para "paragrafoModificado" e será adicionado um novo atributo datanovo-atributo com o valor "Novo valor".

## hasAttribute:

O método hasAttribute é usado para verificar se um elemento possui um atributo específico. Ele recebe um argumento, que é o nome do atributo que queremos verificar. O método retornará true se o atributo existir e false se o atributo não estiver presente. Vejamos um exemplo:

```
// HTML: <a href="https://www.exemplo.com" id="meuLinl</pre>
```

```
const link = document.getElementById('meuLink');
const temHref = link.hasAttribute('href');
console.log(temHref); // Saída: true
```



```
240.3k xp
```





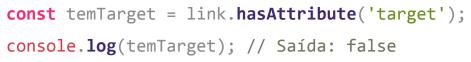
25%

ATIVIDADES 3 DE 9

> DISCORI ALURA

FÓRUM DO CURSO

VOLTAR PARA DASHBOARD



**COPIAR CÓDIGO** 

Neste exemplo, o método hasAttribute foi usado para verificar se o elemento com o ID "meuLink" possui o atributo "href" (que é verdadeiro) e se possui o atributo "target" (que é falso).

## removeAttribute:

O método removeAttribute é usado para remover um atributo específico de um elemento HTML. Ele recebe um argumento, que é o nome do atributo que desejamos remover. Por exemplo:

```
// HTML: <img src="imagem.jpg" alt="Imagem de exemplo'
const imagem = document.getElementById('minhaImagem')
imagem.removeAttribute('alt');</pre>
```



240.3k xp





25%

ATIVIDADES 3 DE 9

> DISCORI ALURA

FÓRUM DO CURSO

VOLTAR PARA DASHBOARD COPIAR CÓDIGO

Após a execução desse código, o atributo alt da imagem será removido, tornando-a menos acessível para pessoas com deficiência visual e fazendo com que o navegador não exiba um texto alternativo quando a imagem não puder ser carregada.

Em resumo, os métodos getAttribute, setAttribute, hasAttribute e removeAttribute são ferramentas poderosas no JavaScript para interagir com atributos de elementos HTML. Eles permitem que os desenvolvedores obtenham, definam, verifiquem e removam atributos conforme necessário, tornando a manipulação do DOM mais eficiente e flexível. Ao utilizar esses métodos de forma adequada, é possível criar experiências web mais dinâmicas e interativas para os usuários.

