



**Olá** pessoal, hoje vou falar sobre um tópico bem básico, mas que muita gente desconhece. Hoje vamos entender as principais diferenças entre `textContent`, `innerText`, `innerHTML`. Todos os 3 métodos estão disponíveis para manipulação do DOM (Document Object Model). Por se tratarem de manipulação do DOM você precisa de pegar o elemento do dom e então chamar essas funções e existem diversas formas de fazer isso, como por exemplo `document.getElementById()` ou `document.querySelector()`, e hoje vamos exemplificar usando `document.getElementById()`.

## Lendo informações do DOM usando textContent, innerText e innerHTML

### innerHTML

Esse com certeza é o mais utilizado para manipulação de DOM, pois existe a mais tempo, mas existem algumas peculiaridades que devemos ficar atentos.

Como se é de esperar, quando usamos o `innerHTML` ele irá retornar todo o texto e o html que existem no elemento, por exemplo:

```
<!DOCTYPE html>
<html>
<head>
  <title>Manipulando DOM com textContent, innerText e innerHTML</title>
</head>
<body>

  <div id="content">
    Texto com <strong>negrito</strong> e texto <span style="display: none;">oculto</span>
  </div>

  <script type="text/javascript">
    const usingInnerHTML = document.getElementById('content').innerHTML;

    console.log(usingInnerHTML);
    // Texto com <strong>negrito</strong> e texto <span style="display: none;">oculto</span>
  </script>
</body>
</html>
```

#### Lendo o DOM usando innerHTML

Como vocês podem perceber, ele retornou todo o html existente na div "content", incluindo o `<strong>` e o `<span>` oculto com css.

### innerText

Quando usamos o `innerText`, diferentemente do `innerHTML`, irá retornar apenas o texto, ignorando todas as tags `HTML` de dentro do elemento, porém `innerText` "entende" o `CSS` do elemento, então retornará apenas textos visíveis.



```
<!DOCTYPE html>
<html>
<head>
  <title>Manipulando DOM com textContent, innerText e innerHTML</title>
</head>
<body>

  <div id="content">
    Texto com <strong>negrito</strong> e texto <span style="display: none;">oculto</span>
  </div>

  <script type="text/javascript">
    const usingInnerText = document.getElementById('content').innerText;

    console.log(usingInnerText);
    // Texto com negrito e texto
  </script>
</body>
</html>
```

Lendo o DOM usando innerText

## textContent

O `textContent`, funciona de forma parecida com o `innerText`, retornando somente o texto e ignorando as tags `HTML`, mas o `textContent` trás todo o conteúdo do texto, incluindo o texto oculto pelo css assim como as quebras de linha ( `\n` ).

```
<!DOCTYPE html>
<html>
<head>
  <title>Manipulando DOM com textContent, innerText e innerHTML</title>
</head>
<body>

  <div id="content">
    Texto com <strong>negrito</strong> e texto <span style="display: none;">oculto</span>
  </div>

  <script type="text/javascript">
    const usingTextContent = document.getElementById('content').textContent;

    console.log(usingTextContent);
    //
    // Texto com negrito e texto oculto
    //
  </script>
</body>
</html>
```

Lendo o DOM usando textContent

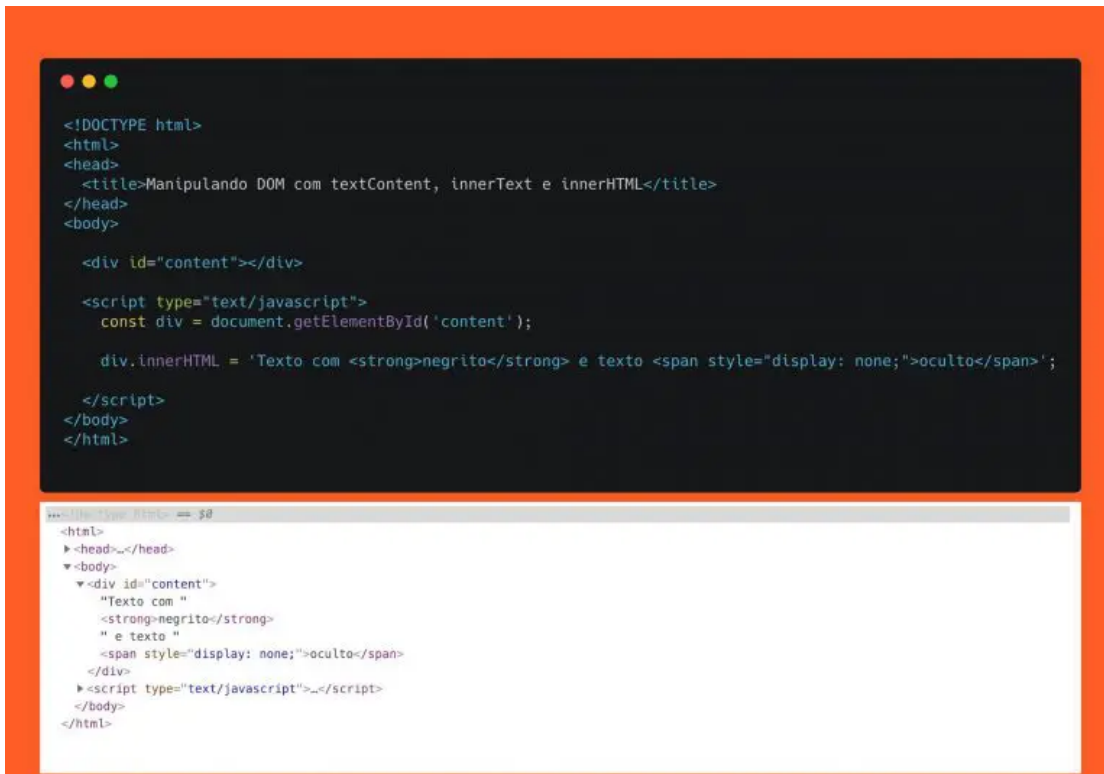
## Escrevendo informações no DOM usando textContent, innerText e innerHTML

Agora vamos ver como essas 3 propriedades se comportam para escrever no DOM.

### innerHTML

Como se é esperado com o `innerHTML`, podemos escrever texto e html no DOM.



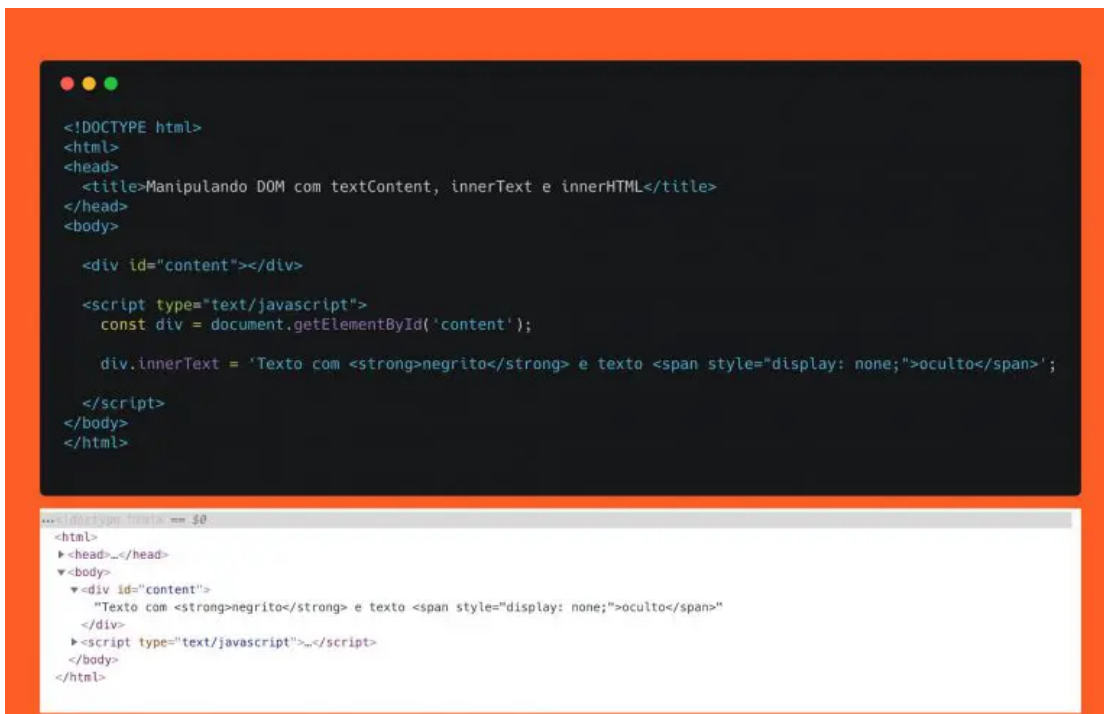


## Escrevendo no DOM usando innerHTML

## innerText e textContent

O `innerText` e `textContent` escrevem o texto no DOM, porém ignoram as tags html, que são escritas como texto também. Na escrita tanto

quanto o `textContent` funcionam de forma similar.



## Escrevendo no DOM com innerText

## Qual deles devo utilizar?

Se o seu intuito for ler ou escrever html dentro de um elemento DOM, você deve usar o **innerHTML** . Se você quer apenas ler o texto, você deve escolher entre o **innerText** e o **textContent** levando em consideração se os textos que não estão visíveis ao usuário deverão ser lidos.

Outra consideração importante a se fazer também durante a escrita de conteúdo no DOM é que se você usar o **innerHTML** e a origem do conteúdo não é de seu controle, você ficará vulnerável à ataques **XSS**.

O textContent não está disponível no IE8, então você deve adicionar um script de forma condicional para que possa usá-lo no IE8, veja como no post do **Eli Grey** sobre o assunto.

## Código completo

O código completo está disponível em nosso [github](#) no [repositório dos posts do blog](#).

```
<!DOCTYPE html>
<html>
<head>
  <title>Manipulando DOM com textContent, innerText e innerHTML</title>
</head>
<body>

  <div id="content">
    Texto com <strong>negrito</strong> e texto <span style="display: none;">oculto</span>
  </div>

  <script type="text/javascript">
    const content = document.getElementById('content');

    const usingInnerHTML = content.innerHTML;
    // Texto com <strong>negrito</strong> e texto <span style="display: none;">oculto</span>

    const usingInnerText = content.innerText;
    // usingInnerText Texto com negrito e texto

    const usingTextContent = content.textContent;
    //
    // Texto com negrito e texto oculto
    //

  </script>
</body>
</html>
```

Lendo o DOM com innerHTML, innerText e textContent

## Obrigado pela leitura ???

Espero que você tenha gostado desse post, creio que agora você irá conseguir tomar decisões mais conscientes ao manipular o **DOM**.

Se você gostou desse post, compartilhe com seus amigos e ajude a espalhar o conhecimento!

Se ainda ficou com dúvidas, por favor deixe nos comentários.

Não se esqueça de nos seguir nas redes sociais, por que você não pode ficar de fora dos novos posts.

1. [Curta nossa página no facebook ?](#)
2. [Não perca as atualizações no twitter ?](#)
3. [Veja as dicas no instagram ?](#)
4. [Siga nossos repositórios no github](#) 📖
5. [Faça parte da nossa lista de desenvolvedores ?](#)

### Hashtags

#javascript #js #es6 #DOM #HTML #innerText #innerHTML #textContent #html #html5 #DOMManipulation

**Quer aprender  
JavaScript?**

