

Template Strings

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

Índice

1. [Sintaxe](#)
2. [Exemplos de uso](#)
3. [Modificando o HTML](#)

1 | Sintaxe

Sintaxe de uma Template String

Texto/String

Este valor se manterá constante.

Código JavaScript

Este código se executará toda vez que o template string for utilizado.

```
JS `Minha variavel vale ${minhaVariavel}`
```

Template String

Se inicia com aspas cruzadas, caso coloque aspas duplas (") ou simples (') o JavaScript entenderá toda a linha como um texto, e a string não funcionará.

Abertura de código

Com estas chaves indicamos que o que está em volta será interpretado como JS, ou seja, se "minhaVariavel" tem um valor de 14, a saída deste template será: **Minha variável vale 14.**

2 | Ejemplos de uso

Exemplo 1

```
const nome= 'Mauro';  
const meuTemplate = `Meu nome é ${nome}`;  
console.log(meuTemplate);
```

O resultado seria:
Meu nome é Mauro



Recomendamos que você teste esse código no seu computador!

Exemplo 2

```
const meuTemplate = `2 * 3 é ${ 2 * 3 }`;  
console.log(meuTemplate);
```

O resultado seria:
2 * 3 é 6

Exemplo 3

```
function soma(var1, var2) {  
    return var1 + var2  
}
```

```
const temp = `O resultado é ${soma(2,2)}`;  
console.log(temp);
```

O resultado seria:
O resultado é 4



Como se pode ver, dentro das chaves pode-se executar qualquer código JavaScript, porém, é uma boa prática que ele seja o mais curto possível.

3 | Modificando o HTML

Conteúdo dinâmico

Já conhecemos funções bem úteis para modificar nosso documento HTML mediante ao JavaScript: **innerText** e **innerHTML**. Agora podemos testar o seguinte fragmento de código em seu computador antes de continuar:

```
const template = `

# 

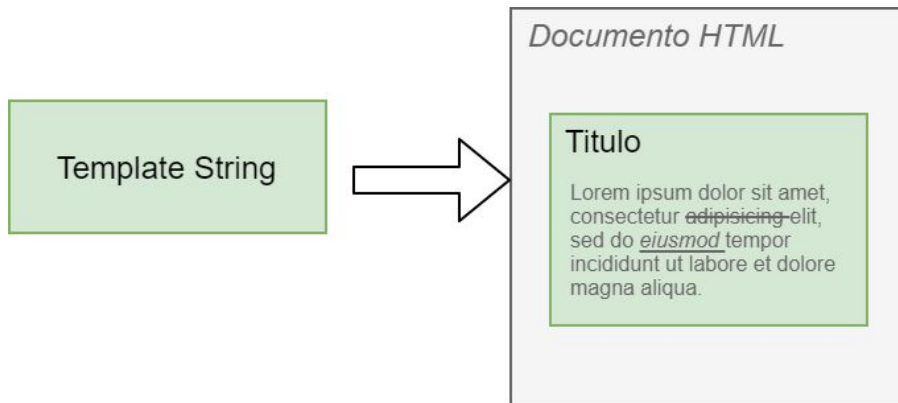

```



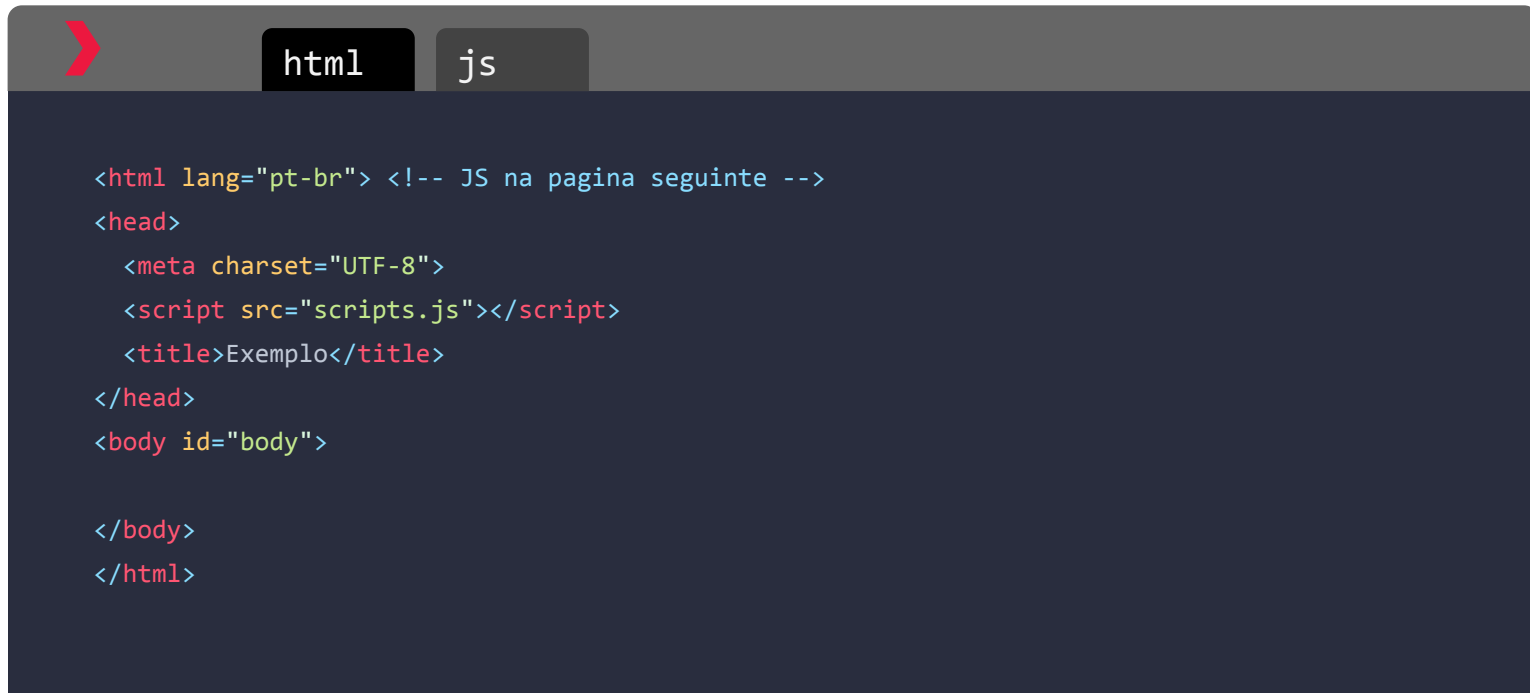
Não se esqueça de vincular o Js ao HTML!

InnerHTML

Diferente de `innerText`, `innerHTML` permite incluir etiquetas no nosso template string e que essas sejam interpretadas corretamente pelo navegador. Isso nos dá a possibilidade de agregar muita dinamicidade em nossa página.



Vamos testar!



```
<html lang="pt-br"> <!-- JS na pagina seguinte -->
<head>
  <meta charset="UTF-8">
  <script src="scripts.js"></script>
  <title>Exemplo</title>
</head>
<body id="body">

</body>
</html>
```



html

js

```
function escreverHTML(titulo, texto) {  
  const body = document.getElementById('body');  
  const meuTemplate = `  
    <h1>${titulo}</h1>  
    <p>${texto}</p>  
  `;  
  body.innerHTML += meuTemplate;  
}  
  
escreverHTML('Olá', 'Isso é um exemplo de template string em HTML.');
```

escreverHTML('É dinâmico', 'Podemos inserir elementos HTML mediante o JavaScript');

escreverHTML('Facilita a programação', 'Evita escrever muito código e reutilizar o que escrevemos.');

escreverHTML('Neste exemplo', 'Utilizamos uma única função para poder escrever 4 vezes em HTML, já imaginou como seria isso sem essa função?');

Template strings

Template strings, template literal, modelos literais, entre outros, são todos nomes que recebe essa funcionalidade e é uma das bases da programação dinâmica na web. De agora em diante, iremos nos aprofundar em diferentes ferramentas e utilidades de JavaScript, e muito frequentemente terminaremos utilizando templates.



Se tiver alguma dúvida, consulte o professor!

DigitalHouse>
Coding School