

JavaScript CreateElement

Summary: in this tutorial, you will learn how to use the JavaScript `document.createElement()` to create a new HTML element and attach it to the DOM tree.

To create an HTML element, you use the `document.createElement()` method:

```
let element = document.createElement(htmlTag);
```

The `document.createElement()` accepts an HTML tag name and returns a new `Node` with the `Element` type.

1) Creating a new div example

Suppose that you have the following HTML document:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JS CreateElement Demo</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

O exemplo a seguir usa o `document.createElement()` para criar um novo `<div>` elemento:

```
let div = document.createElement('div');
```

E adicione um trecho de HTML ao `div` :

```
div.innerHTML = '<p>createElement example</p>';
```

Para anexar o `div` ao documento, você usa o `appendChild()` método:

```
document.body.appendChild(div);
```

Junte tudo:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <title>JS createElement Demo</title>
```

```
</head>
<body>
  <script>
    let div = document.createElement('div');
    div.id = 'content';
    div.innerHTML = '<p>CreateElement example</p>';
    document.body.appendChild(div);
  </script>
</body>
</html>
```

Adicionando um id ao div

Se você quiser adicionar um id a um `div`, defina o `id` atributo do elemento para um valor, assim:

```
let div = document.createElement('div');
div.id = 'content';
div.innerHTML = '<p>CreateElement example</p>';

document.body.appendChild(div);
```

Adicionando uma classe à div

O exemplo a seguir define a classe CSS de uma nova div `note`:

```
let div = document.createElement('div');
div.id = 'content';
div.className = 'note';
div.innerHTML = '<p>createElement example</p>';

document.body.appendChild(div);
```

Adicionando texto a uma div

Para adicionar um pedaço de texto a um `<div>` , você pode usar a `innerHTML` propriedade como no exemplo acima ou criar um novo `Text` nó e anexá-lo ao `div` :

```
// create a new div and set its attributes
let div = document.createElement('div');
div.id = 'content';
div.className = 'note';

// create a new text node and add it to the div
let text = document.createTextNode('createElement example');
div.appendChild(text);

// add div to the document
document.body.appendChild(div);
```

Adicionando um elemento a uma div

Para adicionar um elemento a um `div`, você cria um elemento e o anexa ao `div` usando o `appendChild()` método:

```
let div = document.createElement('div');
div.id = 'content';
div.className = 'note';

// create a new heading and add it to the div
let h2 = document.createElement('h2');
h2.textContent = 'Add h2 element to the div';
div.appendChild(h2);

// add div to the document
document.body.appendChild(div);
```

2 li) Exemplo de criação de novos itens de lista ()

Digamos que você tenha uma lista de itens:

```
<ul id="menu">
  <li>Home</li>
</ul>
```

O código a seguir adiciona dois `li` elementos ao `ul` :

```
let li = document.createElement('li');
li.textContent = 'Products';
```

```
menu.appendChild(li);

li = document.createElement('li');
li.textContent = 'About Us';

// select the ul menu element
const menu = document.querySelector('#menu');
menu.appendChild(li);
```

Saída:

```
<ul id="menu">
  <li>Home</li>
  <li>Products</li>
  <li>About Us</li>
</ul>
```

3) Criando um script exemplo de elemento

Às vezes, você pode querer carregar um arquivo JavaScript dinamicamente. Para fazer isso, você pode usar o `document.createElement()` para criar o `script` elemento e adicioná-lo ao documento.

O exemplo a seguir ilustra como criar um novo `script` elemento e carregar o `/lib.js` arquivo no documento:

```
let script = document.createElement('script');  
script.src = '/lib.js';  
document.body.appendChild(script);
```

Você pode primeiro criar uma nova função auxiliar que carrega um arquivo JavaScript de um URL:

```
function loadJS(url) {  
    let script = document.createElement('script');  
    script.src = url;  
    document.body.appendChild(script);  
}
```

E então use a função auxiliar para carregar o `/lib.js` arquivo:

```
loadJS('/lib.js');
```

Para carregar um arquivo JavaScript de forma assíncrona, defina o `async` atributo do `script` elemento como `true` :

```
function loadJSAsync(url) {  
    let script = document.createElement('script');  
    script.src = url;  
    script.async = true;  
    document.body.appendChild(script);  
}
```

Resumo

- O `document.createElement()` cria um novo elemento HTML.
- A `element.appendChild()` anexa um elemento HTML a um elemento existente.