



JavaScript CreateElement() Method

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>
```

The following example uses the `document.createElement()` to create a new `<div>` element:

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

And add an HTML snippet to the `div` :

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

To attach the `div` to the document, you use the `appendChild()` method:

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

Put it all together:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JavaScript 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>
```

Adding an id to the div

If you want to add an id to a `div`, you set the `id` attribute of the element to a value, like this:

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

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

Adding a class to the div

The following example sets the CSS class of a new div `note` :

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

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

Adding text to a div

To add a piece of text to a `<div>`, you can use the `innerHTML` property as the above example, or create a new `Text` node and append it to the `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);
```

Adding an element to a div

To add an element to a `div`, you create an element and append it to the `div` using the `appendChild()` method:

```
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) Creating new list items (li) example

Let's say you have a list of items:

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

The following code adds two `li` elements to the `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);
```

Output:

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

3) Creating a script element example

Sometimes, you may want to load a JavaScript file dynamically. To do this, you can use the `document.createElement()` to create the `script` element and add it to the document.

The following example illustrates how to create a new `script` element and loads the `/lib.js` file to the document:

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

You can first create a new helper function that loads a JavaScript file from a URL:

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

And then use the helper function to load the `/lib.js` file:

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

To load a JavaScript file asynchronously, you set the `async` attribute of the `script` element to `true` :

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

Summary

- The `document.createElement()` creates a new HTML element.

- The `element.appendChild()` appends an HTML element to an existing element.

Quiz