

the difference between `innerHTML` and `document.createElement`.

The `innerHTML` property : is used to get or set the HTML content of an element. When you set the property of an element, it replaces the existing content of that element with the specified HTML string.

Using `innerHTML` to update the content of an element can be less efficiency because it require reparsing and recreating all DOM nodes inside the element.

`document.createElement`: This method is used to create a new HTML element specified by its tag name. it will create a new `<div>` element. You can then manipulate this element by setting its attributes, adding child elements, or modifying its content.

Unlike `innerHTML` using `createElement` doesnot require reparsing and recreating all DOM nodes inside an element which more efficient. `innerHTML` tutorial, you should use it only when the data comes from a trusted source like a database.

If you set the contents that you have no control over to the `innerHTML`,. the malicious code may be injected and executed

```
//You can new elements to the div element by creating an element and appending it:
let div = document.querySelector('.container');

let p = document.createElement('p');
p.textContent = 'JS DOM';
div.appendChild(p);
```

JS DOM

body 278 x 18

Identify your project's root folder to open source files in Visual Studio

Welcome Elements Console Sources

```
<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body>
    <div class="container"> == $0
      <p>JS DOM</p>
    </div>
    <script src="script.js"></script>
  </body>
</html>
```

JS DOM

body 278 x 18

Identify your project's root folder to open source files in Visual Studio

Welcome Elements Console Sources

```
<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body>
    <div class="container"> == $0
      <p class="child">JS DOM</p>
    </div>
    <script src="script.js"></script>
  </body>
</html>
```

```
//we can also manipulate an element's HTML directly using innerHTML like this
//using innerHTML is cleaner and shorter when you want to add attributes to the element:
let div = document.querySelector('.container');
div.innerHTML += '<p class="child">JS DOM</p>';
```

```
//innerHTML property sets or returns the HTML content (inner HTML) of an element
let text = "texts will be replaced by innerHTML";
document.getElementById("myP").innerHTML = text;
document.getElementById("myDIV").innerHTML = text;
```

the difference between inner and create

texts will be replaced by innerHTML

texts will be replaced by innerHTML