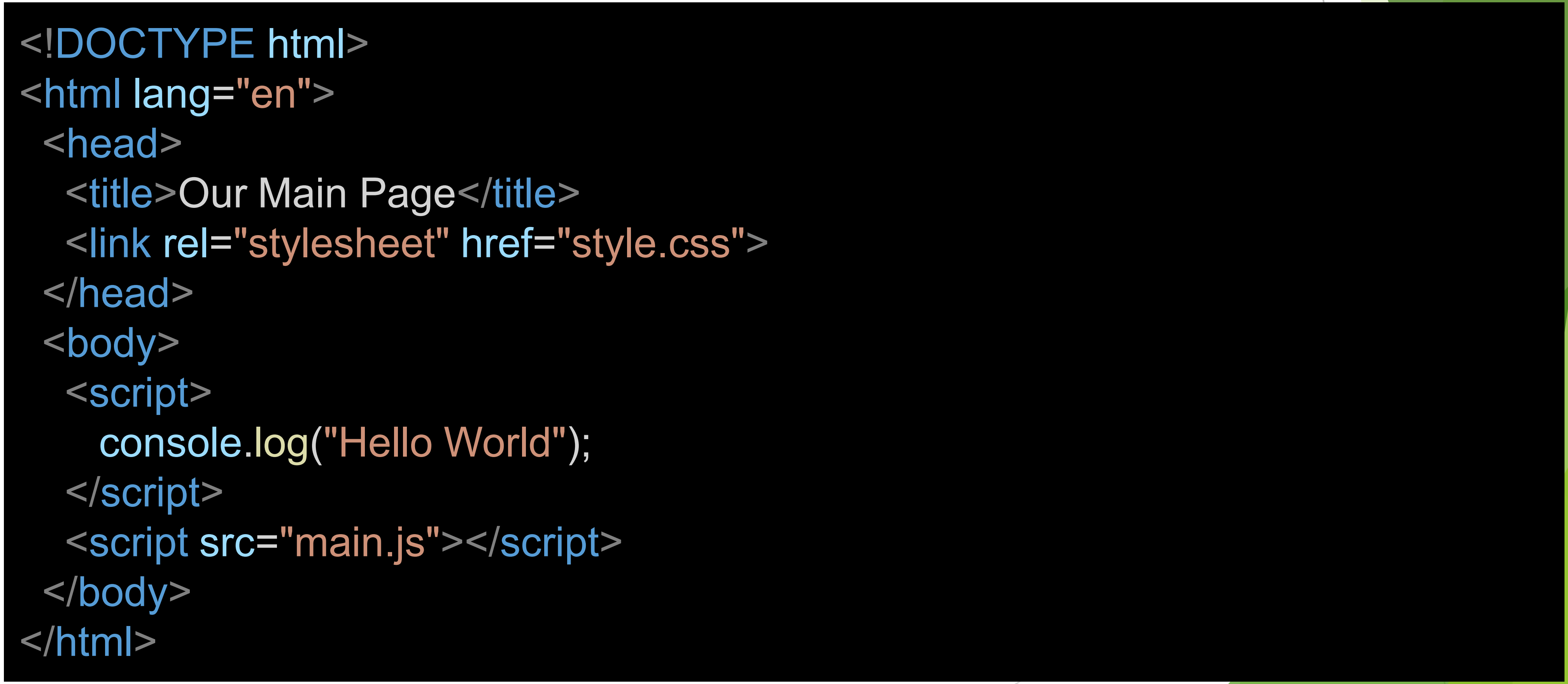
Linking JavaScript File:

- You can use <script> tags to either write JavaScript inside the HTML page or you and link an external file



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Our Main Page</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <script>
      console.log("Hello World");
    </script>
    <script src="main.js"></script>
  </body>
</html>
```

DOCUMENT OBJECT MODEL(DOM):

Selecting elements:

getElementsID:

- The document.getElementById() method returns an Element object that represents an HTML element with an id that matches a specified string otherwise null.
- Unlike the querySelector() method, the getElementById() is only available on the document object, not other elements.

```
const element = document.getElementById(id);
```

- The id is unique within an HTML document. However, HTML is a forgiving language. If the HTML document has multiple elements with the same id, the document.getElementById() method returns the first element it encounters.

```
<html>
 <head>
     <title>JavaScript getElementById() Method</title>
  </head>
  <body>
     <p id="message">A paragraph</p>
  </body>
</html>

<script>
const p = document.getElementById('message');
console.log(p);
</script>
```

createElement():

- To create an HTML element, you use the document.createElement() method.
- The following example uses the document.createElement() to create a new <div> element:

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

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

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

- Add class and id to div

```
div.id = 'content';
div.className = 'note';
```

- Creating new list items li example

```
<ul id="menu">
  <li>Home</li>
</ul>
<script>
// select the ul menu element
const menu = document.querySelector('#menu');
let li = document.createElement('li');
li.textContent = 'Products';
menu.appendChild(li);

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

menu.appendChild(li);
<script/>
```

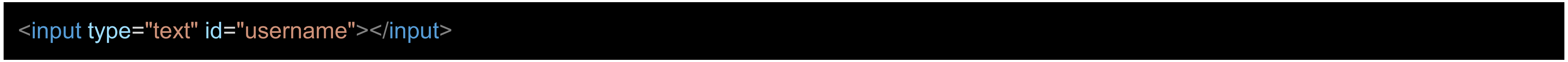
- Output

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

Document Object Model(DOM):

Working with Attributes:

- When the web browser loads an HTML page, it generates the corresponding DOM objects based on the DOM nodes of the document.
- The web browser will generate an HTMLInputElement object.
- For Example:
- The input element has two attributes:
- The type attribute with the value text.
- The id attribute with the value username.
- The generated HTMLInputElement object will have the corresponding properties:
- The input.type with the value text.
- The input.id with the value username.



```
<input type="text" id="username"></input>
```

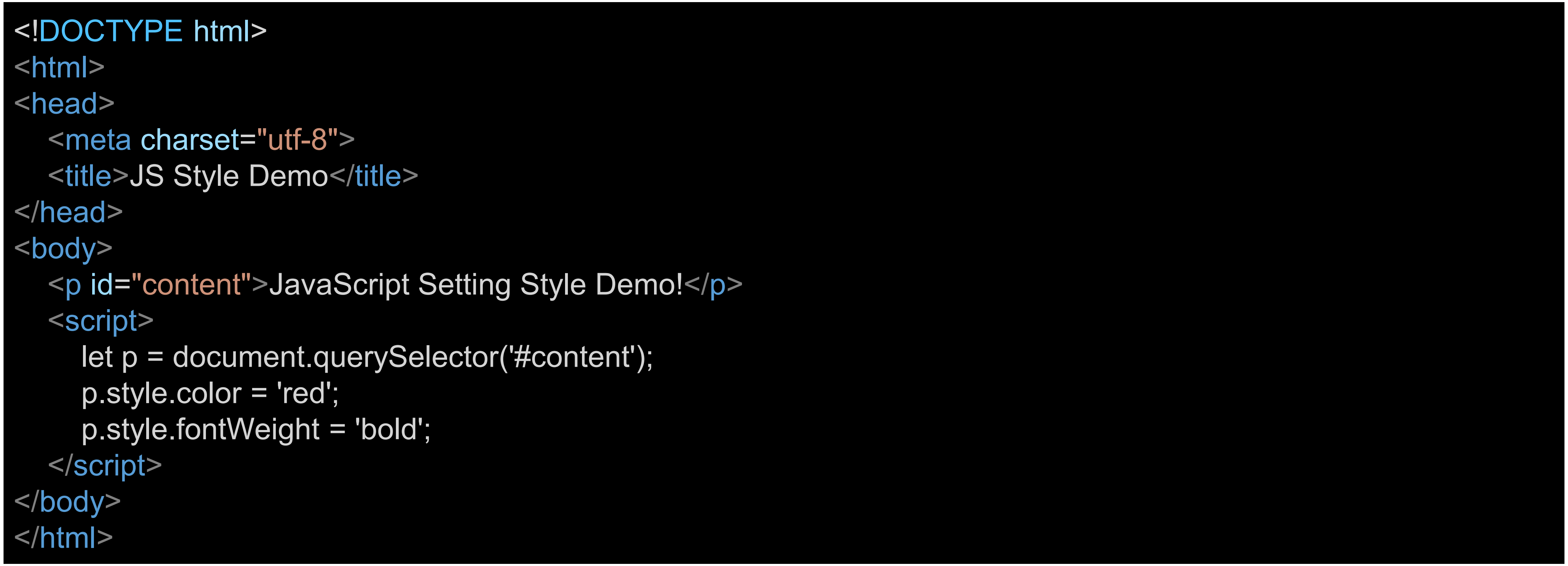
- To access both standard and non-standard attributes, you use the following methods:

element.getAttribute(name) – get the attribute value
element.setAttribute(name, value) – set the value for element.hasAttribute(name) – check for the existence of an attribute
element.removeAttribute(name) – remove the attribute

- The following example shows when the value attribute changes, it reflects in the value property,

```
let input = document.querySelector('#username');

// attribute -> property: OK
input.setAttribute('value','guest');
console.log(input.value);  // guest
```

- The following example uses the style object to set the CSS properties of a paragraph with the id content:

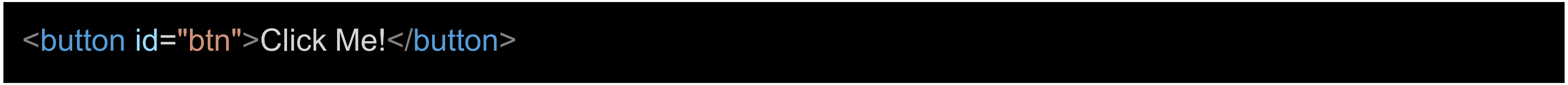


```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JS Style Demo</title>
</head>
<body>
  <p id="content">JavaScript Setting Style Demo!</p>
  <script>
    let p = document.querySelector('#content');
    p.style.color = 'red';
    p.style.fontWeight = 'bold';
  </script>
</body>
</html>
```

Manipulating Element’s Styles:

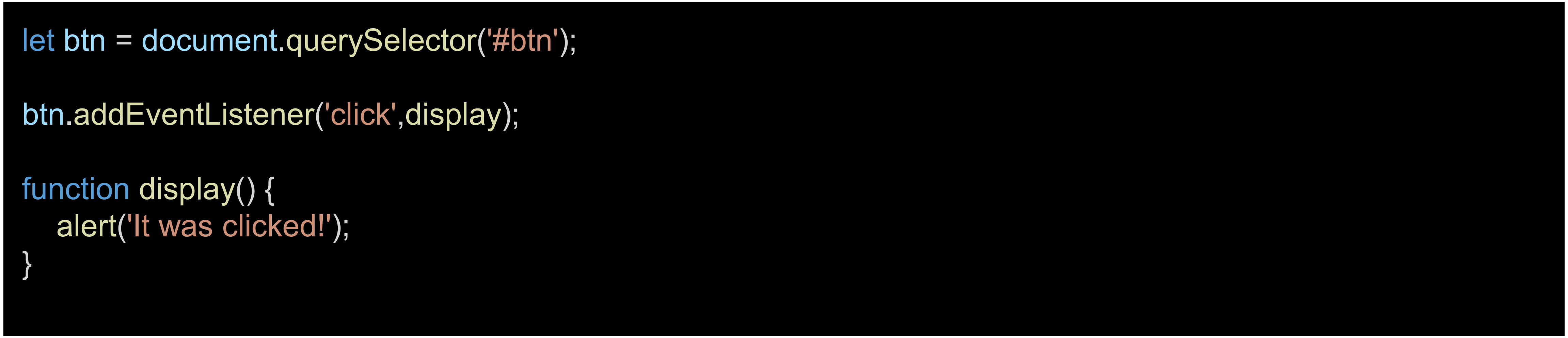
Working with Events:

- An event is an action that occurs in the web browser, which the web browser feedbacks to you so that you can respond to it.
- Each event may have an event handler which is a block of code that will execute when the event occurs.
- An event handler is also known as an event listener. It listens to the event and executes when the event occurs.
- Suppose you have a button with the id btn.



```html
<button id="btn">Click Me!</button>
```

- To define the code that will be executed when the button is clicked, you need to register an event handler using the addEventListener() method.

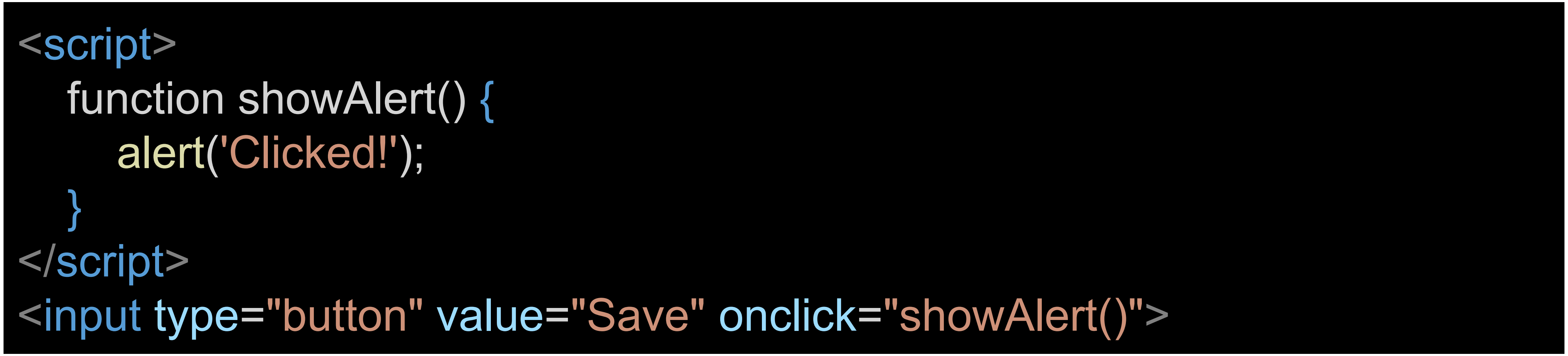


```javascript
let btn = document.querySelector('#btn');

btn.addEventListener('click',display);

function display() {
    alert('It was clicked!');
}
```

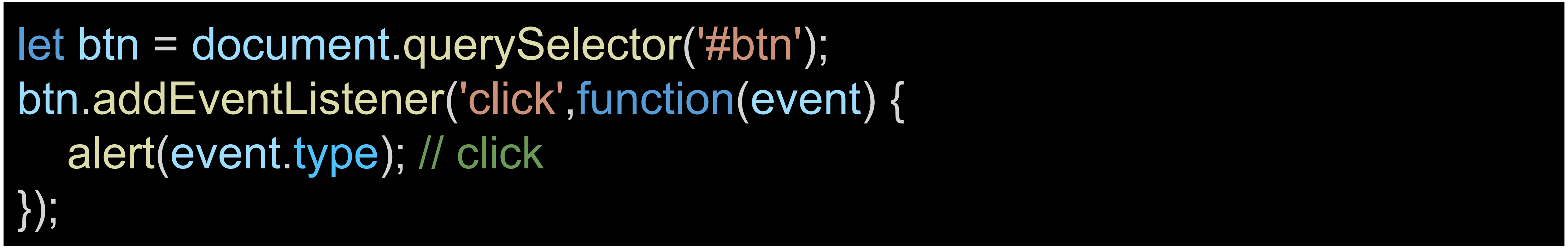
DOCUMENT OBJECT MODEL(DOM):

HTML event handler attributes:

- Event handlers typically have names that begin with on, for example, the event handler for the click event is onclick.



```
<script>
    function showAlert() {
        alert('Clicked!');
    }
</script>
<input type="button" value="Save" onclick="showAlert()">
```



```
let btn = document.querySelector('#btn');
btn.addEventListener('click',function(event) {
    alert(event.type); // click
});
```

Questions?