



Adding JavaScript Code

This lesson will teach you how to add Javascript to build functionality of the HTML form.


When you create an HTML form, pressing a button submits the form by default. Submitting a form reloads the page. Once you reload the page, the data entered in the textfield is lost. This is what we will prevent now.

Output
JavaScript
HTML
CSS (SCSS)

```
function helloListener( event ) {  
    event.preventDefault();  
    console.log( 'button pressed' );  
}  
  
const helloButton = document.querySelector( '.js-hello' );  
helloButton.addEventListener( 'click', helloListener );
```



Console

 Clear

First, we created a `helloListener` function. This function prevents the default action of the event, which is the submission of the form.

The second line in the function creates a console log that appears in the developer tools of your browser. More on this later.

The last line attaches the `helloListener` function to the button. We tend to use `js-` prefixed classes to refer to elements in the Document Object Model, also

known as the DOM. The `document.querySelector` function takes a selector

string, in this case a class name, and locates the node in the DOM that has this class. Check out the HTML tab, you can see the same class in the class list of the button.

Once we located the `.js-hello` button, we can add an event listener function to it. This function takes an event, and can perform any JavaScript action ranging from manipulating the contents you can see on screen to calling a service or an API on the web, saving your data in a persistent storage.

If you will work on your computer, you will face one problem:

You will not have access to Javascript code in the HTML file. You will have to add an HTML tag

```
<script src="yourjsfilename.js"> </script>
```

at the bottom of the body.

Developer Tools

Each browser has developer tools. For example, in Google Chrome right clicking on your website anywhere inside the browser window, and selecting `Inspect` from the context menu, you will find yourself inside the developer tools. Find the `Console` tab.

Assuming you have clicked on the button, you can find the following there:

```
button pressed
```

Your console logs will also be displayed underneath your code.

In the Developer Tools, You can execute any JavaScript expression by writing it after the `>` prompt:

```
button pressed
> 2+2
4
> helloButton
<button class="btn-large js-hello">Greet</button>
```

The `helloButton` variable stores a DOM node, fully accessible in JavaScript.