

Quick word on `<forms>`

## What if we wrapped everything in a `<form>` element?

In the previous example we did not wrap the `<select>`, `<button>` and `<div>` elements inside a `<form>`. Why? Forms are a beast of their own. What I mean by this is that they are subject to certain rules.

What kind of rules?

For example, when you click on a submit button (or an input element with `type="submit"`), then the browser will automatically validate all of the `<input>` elements in the form and submit it to the action URL if there aren't any validation problems. The result of this when you have not specified an action URL is that the page will refresh.

Therefore, in our previous example, **IF we were to wrap** all of our elements inside a `<form>` tag, then eventually when the form is submitted, the entire browser will refresh. You will only see the message appear in the `<div>` tag momentarily, before everything resets itself.

**Bottom line: when a submit event is fired within a form, the browser will automatically send that form data to a server, and the browser will refresh the page .**

## How do we stop this unwanted behavior?

The only way to avoid this behavior is to make the form submit via JavaScript or AJAX.

Let me show you a quick example of using JavaScript, and the `preventDefault()` method. In case you are wondering, the `preventDefault()` method is given to us by the browser (specifically, it is a Web API given to us by the DOM). As the name suggests, using this method tells the browser that the default action (in our case the page refresh), should not happen.

Just remember, when we listen for an event, we get the event object passed to us automatically. In the below code we've put this object into a variable we've named `e`. This is where we get our `preventDefault()` method from.

Ready?

Ok, here's the code.

```

1  <html>
2    <body>
3      <form>
4        This is a random input box <input type="text">
5        <button type="submit">Submit</button>
6      </form>
7      <div></div>
8    </body>
9    <script>
10     // we can access our form directly on the document object
11     let myForm = document.forms[0];
12     // lets now use JavaScript to handle the form submission
13     myForm.addEventListener('submit', (e) => {
14       // you need to prevent the default behavior of a browser refresh
15       e.preventDefault();
16       // lets grab our div tag and insert a message into it
17       let messageDiv = document.getElementsByTagName('div')[0];
18       messageDiv.innerText = "We have prevented the page from reloading";
19     })
20   </script>
21 </html>

```

And voila!

We have the result of having our message inserted into our `<div>` tag.



Don't worry if you are a little confused ... by the end of this course this will be a piece of cake for you ;)

See you in the next lectures.

