

Linking a JavaScript File

Introduction

In this lecture, we will learn how to write JavaScript code in a separate file and run it in the browser by linking it to an HTML file.

Writing JavaScript in HTML

To start, you always need an HTML document because your JavaScript needs to be attached to it. In HTML, there is a **script** tag where you can write JavaScript. Place it right beneath the style tag.

If you are new to HTML, there is a short crash course on HTML and CSS in section five of this course. You can take that to learn the basics.

Let us create a script tag and write JavaScript inside it. In the first video, we wrote JavaScript in the console, but real development is done in files.

For now, write the same code as in the first video: declare a variable and show an alert if the condition is met.

javascript Code Sample

```
let js = 'amazing';
if (js === 'amazing') alert('JavaScript is fun!');
```

In JavaScript, it is common to add a semicolon at the end of each line. While not mandatory, it is considered good practice and makes the code cleaner.

To execute this code and show the alert window, open **index.html** in the browser. The alert will appear, confirming that your JavaScript is working.

Using the Browser Console

Let us try a math operation as in the previous lecture.

javascript Code Sample

```
40 + 8 + 23 - 10;
```

If you reload the page, you will notice that the result does not appear in the alert or on the page, nor in the console. This is because JavaScript does not output results to the console by default when running from a script.

To display results in the console, you need to explicitly output them using the **console.log** function.

javascript Code Sample

```
console.log(40 + 8 + 23 - 10);
```

After saving and reloading, you will see the result printed in the console, along with the line number from **index.html** where the computation was performed.

Recap: Console vs Script Tag

In the previous lecture, you were already in the console, so there was no need for **console.log** or a script tag. Now, since you are writing in a script, you must use **console.log** to output to the developer console.

Moving JavaScript to an External File

Currently, the script is inline in the HTML file. While this means you do not have to load another file, it is not ideal for separating website content from JavaScript logic. Usually, external JavaScript files are used.

Create a new file named **script.js** in the same folder as **index.html**. This is a standard name for a JavaScript script.

Copy your JavaScript code from the script tag and paste it into **script.js**. You can now remove the script tag from your HTML file.

Linking the JavaScript File to HTML

To link the JavaScript file to your HTML, add a script tag at the end of the body tag, but this time use the **src** attribute to specify the file name.

html Code Sample

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

If **script.js** is in the same folder as **index.html**, you only need to write the file name. After saving and reloading, your JavaScript will run as before.

If you do not see the alert window or output, make sure that **script.js** is in the same folder as **index.html**, that the script tag is correctly written, and that there are no errors in your JavaScript code.

Conclusion

Now you have a clean separation between your code logic and presentation. You know how to link a JavaScript file to an HTML file and run it in the browser. You are now ready to start learning the JavaScript language in depth.

Key Takeaways

- Download the starter code from the provided GitHub repository and extract it to begin the course.
- Use the starter folder as your project folder in VS Code and keep the final folder for reference.
- Write JavaScript code inside a script tag in the HTML file, and use **console.log** to output results to the browser console.
- For better code organization, move JavaScript code to an external **.js** file and link it to the HTML using the **script** tag with the **src** attribute.