JavaScript's role in Web Development

What is JavaScript (JS)?

- JavaScript (JS) is the #1 programming language for a web developer to learn.
- JS is used by millions of websites and is required for nearly all developer jobs.
- On the Frontend, JS "sees" web pages as a hierarchical collection of objects: the **Document Object Model (DOM)**.
- On the Backend, **Node.js** and **React.js** work with a **server** object and databases.

What can Javascript do for Websites?

- **Add interactivity**

JavaScript enables interactivity, which empowers the user. When you click a button and something happens, that's probably JavaScript in action.

Manipulate the DOM

JS can get any html element and do stuff, like:

- change an element's content or modify an element's CSS properties:
- retrieve values from form objects
- perform calculations and output the result
- When fresh data appears without the whole page reloading, that's JS.

JavaScript is written...

- between a pair of tags, in either the head or body of an **html** file.
- in a **.js** file, which is imported into either the head or body of an html page.

Per Wikipedia:

```
JavaScript (JS), is a programming language that is one of the core
technologies of the World Wide Web, alongside HTML and CSS. Over 97% of
websites use JavaScript on the client side for web page behavior, often
incorporating third-party libraries. All major web browsers have a
dedicated JavaScript engine to execute the code on users' devices...

JavaScript engines were originally used only in web browsers, but are now
```

```
core components of some servers and a variety of applications. The most
popular runtime system for this usage is Node.js ...

Although Java and JavaScript are similar in name, syntax, and respective
standard libraries, the two languages are distinct and differ greatly in
design.
```

1. Launch your code editor (Visual Studio Code, Sublime Text, etc.). If you are in a Noble Desktop class, launch Visual Studio Code.
2. Navigate to the JavaScript-Fundamentals folder.
3. Open the 00-Introduction folder and from in there, open the file **00.01-Introduction-to-JavaScript.html**.

script tag

When written directly in an html page, JS is wrapped in a script tag, which goes right before the closing body tag. Putting the JS code at the end lets the web page fully load before any JS "goes looking" for any particular element to manipulate.

4. Add a pair of script tags right before the closing body tag. Separate the script tags and skip a line:

```
    <script>
    </script>
</body>
```

output

Programs produce output – be it a single-line greeting or a complete, interactive application. Output is also necessary at each step in the coding process, as it provides feedback that tells if the code written so far is working—or if it's buggy (has errors).

alert()

alert() is a JS method. Think of methods as the verbs of programming languages. They perform actions. In this case, the action is to pop-up a dialog box that displays the string (text) which was passed to the parentheses.

```
    alert('Hello World');
```

string

In JS and other programming languages, plain text is known as a string. Strings go in quotes, either single or double.

5. Inside the script tag, add an alert with the string: "Hello World"

```
<script>
    alert('Hello World');
</script>
```

6. Save the file and open it in Chrome. You should see a dialog that says *Hello World*.
7. Close the alert, but leave the html page open in the browser. We'll be reloading the page often as we proceed.

console.log()

Running Javascript in the Browser Console • You do not need a DOM, that is to say, a web page, in order to write and run Javascript. • You can play with JS directly in the Console, as many developers do—no html page required. • To output content to the Console, use console.log(). In the Chrome browser, type Command-Option-I to open Developer Tools. Add this command:

Now, we'll switch to the console.log() method, which is what we'll be using from now on. console.log() also outputs whatever is in its parentheses, but its output goes to the Console.

8. Use the console.log() method to output another message. Strings (text) can go in either double or single quotes, so switch to double quotes:

```
alert('Hello World');
console.log("Hello from the Console");
```

9. Save the file and reload the browser. You get the alert again, but there's more:
10. Right-click the page and choose Inspect; then click the Console tab. It should say: Hello from the Console

Global Window Object

The *global window object* is the top-level object in JavaScript, corresponding to the browser window itself.

- *alert()* is a method of the window
- *console* is an child object of the *window* object.
- *log()* is a method of the *console* object. Since the window object is top-level, it is assumed, and therefore can be omitted from the syntax. But we can also add them:

11. Put *window* in front of the alert and console keywords and run it again. It still works:

```
window.alert('Hello World');
window.console.log("Hello from the Console");
```

semi-colon

Notice the semi-colons (😉 which mark the end of each line of code. Semi-colons are optional, but recommended. We'll be using them.

comment

Comments explain what the code is doing. Comments are for us humans and, as such, are ignored by JS when the program is run. Comments have no effect on how a program runs, but they do make a program easier to read and understand. We recommend you comment your code.

single-line comment

A double slash at the start of a line turns the whole line into a comment.

12. Add a couple of one-line comments to start the script:

```
// alert() and console.log() output what's in parentheses
// window. is optional, since it is understood
window.alert('Hello World');
window.console.log("Hello from the Console");
```

in-line comment

If a comment is short enough, it can go on the same line as the code to which it refers.

13. Add an in-line comment:

```
// alert() and console.log() output what's in parentheses
window.alert('Hello World');
window.console.log("Hello from the Console");
```

commenting-out code

Comments are used to deactivate code without having to delete it. A double slash at the start of a line comments it out.

14. We don't want to keep getting that pop-up, so comment-out the alert.

```
// window.alert('Hello World');
window.console.log("Hello from the Console");
```

We could have just deleted the alert, but by commenting it out instead, we keep an example for study and review purposes.

multi-line comments

If a comment runs more than one line, you can wrap all in /* ... */.

15. Replace the single-line comment at the top of the script with a multi-line comment:

```
/*
2 ways to output JS:
1.) alert() for a pop-up in the browser
2.) console.log() for output to the console
*/
// window.alert('Hello World');
window.console.log("Hello from the Console");
```