Advanced Web Development

CS163

Basic Concepts of Javascript

What is JavaScript?

JavaScript ("JS" for short) is a full-fledged dynamic programming language that can add interactivity to a website. It was invented by Brendan Eich (co-founder of the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation).

JavaScript is versatile and beginner-friendly. With more experience, you'll be able to create games, animated 2D and 3D graphics, comprehensive database-driven apps, and much more!

JavaScript itself is relatively compact, yet very flexible. Developers have written a variety of tools on top of the core JavaScript language, unlocking a vast amount of functionality with minimum effort. These include:

- Browser Application Programming Interfaces (APIs) built into web browsers, providing functionality such as dynamically creating HTML and setting CSS styles; collecting and manipulating a video stream from a user's webcam, or generating 3D graphics and audio samples.

- Third-party APIs that allow developers to incorporate functionality in sites from other content providers, such as Twitter or Facebook.

- Third-party frameworks and libraries that you can apply to HTML to accelerate the work of building sites and applications.

A Hello world! example

JavaScript is one of the most popular modern web technologies! As your JavaScript skills grow, your websites will enter a new dimension of power and creativity.

However, getting comfortable with JavaScript is more challenging than getting comfortable with HTML and CSS. You may have to start small, and progress gradually. To begin, let's examine how to add JavaScript to your page for creating a Hello world! example.

1. Go to your test site and create a new folder named scripts. Within the scripts folder, create a new file called main.js, and save it.

2. In your index.html file, enter this code on a new line, just before the closing </body> tag:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>JavaScript Hello World Example</title>
   <script src="scripts\main.js"></script>
</head>
<body>

</body>
</html>
```

3. This is doing the same job as the <link> element for CSS. It applies the JavaScript to the page, so it can have an effect on the HTML (along with the CSS, and anything else on the page).

4. Add this code to the main.js file:

```
alert(‘Hello World’);
```

Variables

Variables are containers that store values. You start by declaring a variable with the var or the let keyword, followed by the name you give to the variable:

```
let myVariable;
```

After declaring a variable, you can give it a value:

```
myVariable = 'Bob';
```

Also, you can do both these operations on the same line:

```
let myVariable = 'Bob';
```

After assigning a value to a variable, you can change it later in the code:

```
let myVariable = 'Bob';
myVariable = 'Steve';
```

Data types

Variable	Explanation	Example
String	This is a sequence of text known as a string. To signify that the value is a string, enclose it in single quote marks.	let myVariable = 'Bob';
Number	This is a number. Numbers don't have quotes around them.	let myVariable = 10;
Boolean	This is a True/False value. The words true and false are special keywords that don't need quote marks.	let myVariable = true;
Array	This is a structure that allows you to store multiple values in a single reference.	let myVariable = [1,'Bob','Steve',10]; Refer to each member of the array like this: myVariable[0], myVariable[1], etc.
Object	This can be anything. Everything in JavaScript is an object and can be stored in a variable. Keep this in mind as you learn.	let myVariable = document.querySelector('h1'); All of the above examples too.

Comments

```
/*
Everything in between is a comment.
*/


// This is a comment
```

Operators

Operator	Explanation	Symbol(s)	Example

Addition	Add two numbers together or combine two strings.	+	6 + 9;
			'Hello ' + 'world!';

Subtraction, Multiplication, Division	These do what you'd expect them to do in basic math.	-, *, /	9 - 3;
			8 * 2; // multiply in JS is an asterisk
			9 / 3;

Assignment	As you've seen already: this assigns a value to a variable.	=	let myVariable = 'Bob';

Equality	This performs a test to see if two values are equal. It returns a true/false (Boolean) result.	===	let myVariable = 3;
			myVariable === 4;

Not, Does-not-equal	This returns the logically opposite value of what it precedes. It turns a true into a false, etc.. When it is used alongside the Equality operator, the negation operator tests whether two values are *not* equal.	!, !==	For "Not", the basic expression is true, but the comparison returns false because we negate it:
			let myVariable = 3;
			!(myVariable === 3);
			"Does-not-equal" gives basically the same result with different syntax. Here we are testing "is myVariable NOT equal to 3". This returns false because myVariable IS equal to 3:
			let myVariable = 3;
			myVariable !== 3;

Conditionals

```
let iceCream = 'chocolate';
if(iceCream === 'chocolate') {
  alert('Yay, I love chocolate ice cream!');
} else {
  alert('Awwww, but chocolate is my favorite...');
}
```

Functions

```
let myVariable = document.querySelector('h1');
```

```
alert('hello!');
```

You can also define your own functions. In the next example, we create a simple function which takes two numbers as arguments and multiplies them:

```
function multiply(num1,num2) {
  let result = num1 * num2;
  return result;
}
```

```
multiply(4, 7);
```

Events

Real interactivity on a website requires event handlers. These are code structures that listen for activity in the browser, and run code in response. The most obvious example is handling the click event, which is fired by the browser when you click on something with your mouse. To demonstrate this, enter the following into your console, then click on the current webpage:

```
document.querySelector('html').onclick = function() {
   alert('Ouch! Stop poking me!');
}
```

Adding an image changer

In this section, you will learn how to use JavaScript and DOM API features to alternate the display of one of two images. This change will happen as a user clicks the displayed image.

1. Choose an image you want to feature on your example site. Ideally, the image will be the same size as the image you added previously, or as close as possible.
2. Save this image in your images folder.
3. Rename the image firefox2.png.
4. Add the JavaScript below to your main.js file.

```
let myImage = document.querySelector('img');

myImage.onclick = function() {
    let mySrc = myImage.getAttribute('src');
    if(mySrc === 'images/firefox-icon.png') {
      myImage.setAttribute('src','images/firefox2.png');
    } else {
      myImage.setAttribute('src','images/firefox-icon.png');
    }
}
```

5. Save all files and load index.html in the browser. Now when you click the image, it should change to the other one.