Introduction

Welcome to Introduction to JavaScript!

JavaScript is one of the easier programming languages to learn and master. However, it's used mainly in conjunction with CSS and HTML.

So, why should you learn JavaScript? In a nutshell, if you want to do Web development work, you need to know it. The vast majority of Webpages out there right now contain some JavaScript code. With our other two languages, CSS and HTML, it's essential for Web developers.

Meet the Instructor

My name is Alan Simpson, and I'll be your instructor. My interest in the Internet began in the early 1990s when the public first started getting excited about the World Wide Web. Like most people, I was hesitant to learn HTML, but soon realized the importance of having the right tools when creating in the digital space.

Thankfully, I took the plunge and learned HTML, CSS, and JavaScript languages. And I’m glad I did, because those three languages not only rule the Internet, but also mobile app and desktop app development.

These days, I spend most of my time teaching online courses and developing Web applications for a large organization. My goal is to help aspiring developers like you learn three of the most important tools for digital creativity: CSS, HTML, and JavaScript.

Who is this Course For?

Outside of having some basic file and folder skills for your operating system, before attempting this course, you should ideally already be quite familiar with CSS and HTML. The more you know about CSS and HTML, the easier it is to understand what JavaScript is all about and how you might use it in your own work.

Recommended Courses

Creating Webpages (https://www.ed2go.com/courses/computer-science/programming/ilc/creating-web-pages)

If you are unfamiliar with HTML and CSS, this course would be a good start.

View Course (https://www.ed2go.com/courses/computer-science/programming/ilc/creating-web-pages)

HTML and CSS Series (https://www.ed2go.com/courses/computer-science/programming/bundle/html-and-css-series)

Completing the introductory and intermediate CSS and HTML courses in this series would be an even better start.

View Course (https://www.ed2go.com/courses/computer-science/programming/bundle/html-and-css-series)

But don't let these recommended requirements scare you off if you're ready to dive into the programming languages. You can learn this, just as millions of others have. This course will start with the basics and help you work your way towards becoming fluent in JavaScript.

Regardless of your experience level, as you complete this course, you'll discover that JavaScript is not as scary as it sounds. You'll also be proud of what you've accomplished and better armed to succeed in this crazy, high-tech digital world.

The easiest way to understand what JavaScript does is to think about it in relation to HTML and CSS. A lot of this is probably review for you, but it's helpful to look at the big picture to understand how JavaScript will fit on the Webpages you'll be creating.

So let's start with the most basic definition of what JavaScript *is*.

Trivial Example

In this lesson, you'll write and test your first JavaScript code. We'll use a simple example to get going. In web development, we call this a *trivial example*.

Remember

You have to start with the basics and work your way up to the more advanced things. That's the only way to really understand the language and develop the kind of fluency that allows the pros to be so creative.

Chapter 1: What Is JavaScript?

JavaScript is Code

When you're creating a webpage, you're bringing two things together:

- ***Content*** - The material you want to display on the webpage. This is what you want visitors to see and interact with.
- **Code** - The programming language that works behind the scenes to determine how and where the content appears on the webpage.The code is the stuff for the computer to read and process, not for the person who is viewing the page.

HTML, CSS, and JavaScript are all kinds of code.

HTML is a *markup language* that tells the browser what each element *is*. You'd use HTML code for putting text, links, and pictures on Webpages. The CSS code defines how the content *looks*. It's a styling language that applies the styles you choose—font, font color, formatting, and so forth—to all the pages that are ruled by that style. And JavaScript controls how the page behaves or reacts when users click something on the page. In other words, the JavaScript code determines how things *behave*. So whether you're writing HTML tags, CSS style rules, or JavaScript, you're writing code.

Now let's talks about how JavaScript works.

JavaScript is a Scripting Language

When you use JavaScript, you write the code into the file you use to create a webpage. Then you run it by simply opening the page in a browser. So, you're not creating a separate, *stand-alone* program. Instead, JavaScript is something that runs within your webpage, alongside your HTML code and CSS code. That's because JavaScript is something we call a *scripting language* (also called an *interpretive language*).

With scripting languages, you're only using the language to fine-tune or enhance things you've already created using some other language, such as CSS or HTML. As a result, it's pretty easy to use JavaScript, and development time tends to be short (and inexpensive). That's probably why it's so popular on the Web!

In contrast, there are other programming languages called *compiled languages*. With a compiled language, you write your code in a file, but you can't run that file directly. You have to *compile* the code in that file first to create a second executable file. That *executable* file is a stand-alone file with a different filename extension (such as .exe in Windows or .app in Mac OS).

Now you have a sense of what JavaScript is, what it does, and how it fits on a webpage.

JavaScript Is Not Java

Before we move on, we'd like to draw your attention to an important distinction. There's actually a lot of confusion on this point. But let's settle this right now: JavaScript and Java are not the same thing. In fact, Java and JavaScript are two entirely separate languages serving two entirely different purposes. They have almost nothing in common at all.

JavaScript is a scripting language that you use in conjunction with other languages and that executes automatically when you use a browser. In contrast, Java is a compiled language used for creating all kinds of software—so it has nothing to do with HTML, CSS, the Web, or webpages

Google's Android operating system, found on countless Android phones and tables, is largely written in Java. Some popular apps for Windows, iOS, and Mac computers are written in Java, as are many of electronic components in modern cars and TVs.

Java has a wide reach outside the realm of web development, but inside the world of web development, it's all about JavaScript. In fact, there are exactly three client-side languages that all modern web browsers can execute. And those are CSS, HTML, and JavaScript (not Java).

Bottom line: Java is not JavaScript. Knowledge of Java won't help you create better webpages. But knowledge of JavaScript will help you create better webpages.

Note

In case you're wondering, ECMA originally stood for European Computer Manufacturers Association. But now that they are a global company, they say the acronym doesn't really stand for anything.

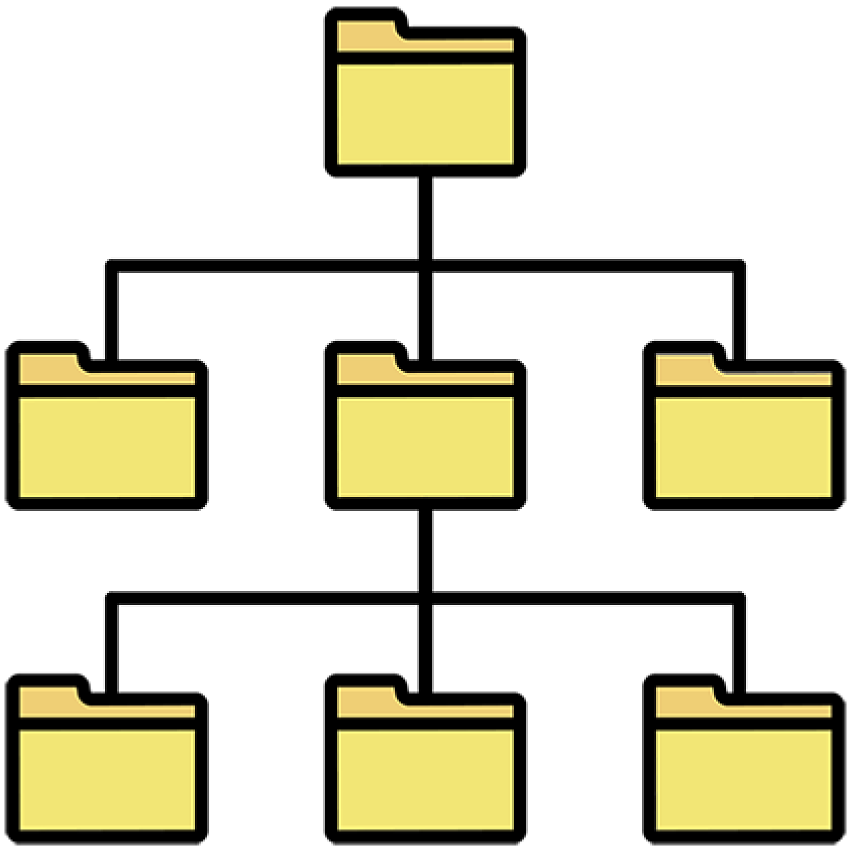
Now let's get geared up to start learning JavaScript.

Chapter 2: Are You Ready to Work with JavaScript?

Storing Your Work

After all you learned in the last chapter, you're probably excited to start learning how to use JavaScript. Not so fast! First, you need to make sure you're set up to work as efficiently as possible by getting organized.

As you likely know, when you create webpages, you'll end up with a lot of files that need to go together. Rather than having an endless sea of files, it’s best to create file folders—and even a file hierarchy structure—to group files by project.



Since we'll be working with webpages in this course, we suggest you create a new folder, just for this course: *Intro to JavaScript*. We'll refer to this folder throughout the course, so we recommend you use that name as well.

Exactly where you store that folder is entirely up to you. Just make sure it's easy to get to. If you want to create the new folder on your desktop, make sure you can see the desktop. Otherwise, open the storage location in which you want to create the new folder.

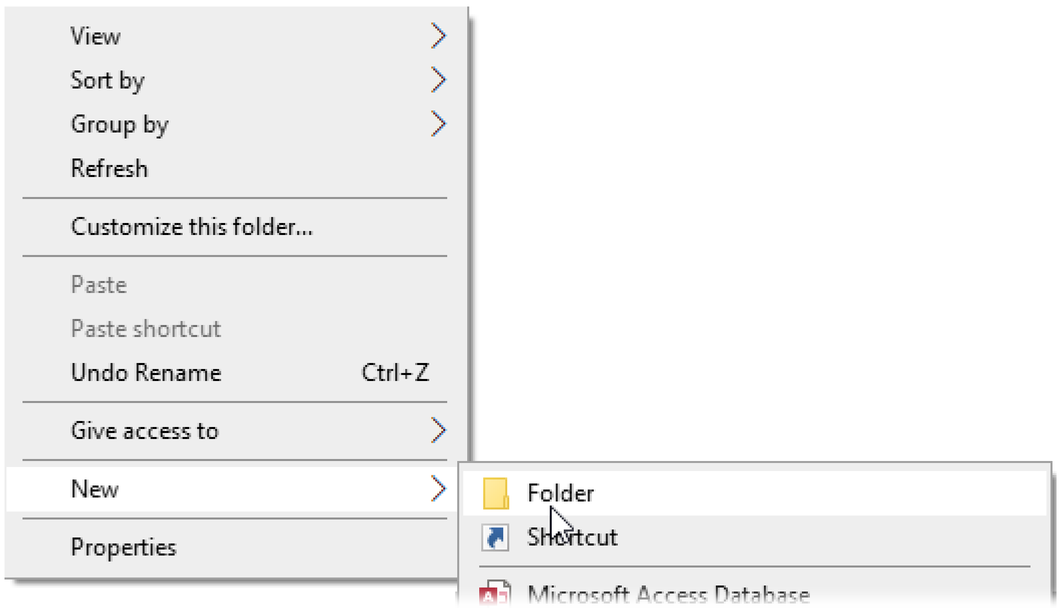
Creating a Course Folder

Your course folder is like every other folder on your hard drive, so you can create it through the usual means for your operating system.

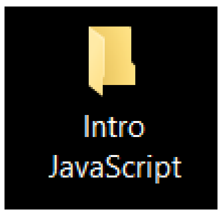
Whether you are adding the folder to your desktop or another location on your computer, here are some basic instructions for creating a new folder, in case you need them:

Here are the Steps

1. Right-click some empty space between icons, choose **New** > **Folder**.



2. Type *Intro JavaScript* as the folder name and press **ENTER**.



Windows OS

[Microsoft Support: Create a new folder (https://support.office.com/en-us/article/create-a-new-folder-cbbfb6f5-59dd-4e5d-95f6-a12577952e17)](https://support.office.com/en-us/article/create-a-new-folder-cbbfb6f5-59dd-4e5d-95f6-a12577952e17)

MacOS

[Apple Support: Organize files in folder on Mac (https://support.apple.com/guide/mac-help/organize-files-using-folders-mh26885/mac)](https://support.apple.com/guide/mac-help/organize-files-using-folders-mh26885/mac)

When you're done, you should see a standard folder icon for your operating system (OS). The exact size and appearance of that icon depends on the brand and version of your OS and your current **View** setting. But it's just a folder like any other, so its icon will look like any other folder icon in your current view.

Building Webpages

So now you've got a place to keep the files you'll be creating in this course. Now, you need to actually create your first page file so that you have something to use when we start working with JavaScript.

Creating Pages

Because JavaScript is used in webpages, you'll be working in webpages throughout this course. The way you create those webpages is exactly the same way you create webpages that don't contain any JavaScript code. Hopefully you've done that before, so these steps will mainly serve as review.

You'll use a plain text editor or whatever you've used in the past to create webpages. Each page should contain a DOCTYPE and the minimum HTML tags that most browsers expect to find in a page. In this course, we'll be using HTML5 since that is the prevailing version.

Let's go ahead and start by creating a new webpage.

Here are the Steps

1. Open whatever program you normally use to create and edit Webpages (Notepad, Notepad + + , TextEdit, TextWrangler, Komodo Edit, Dreamweaver, or whatever is your preference).

Tip

If you're using an iCloud-enabled version of TextEdit in Mac OS and want to put the page in the folder on your hard drive, choose **On My Mac** from the top of the dialog box that opens, and click **New Document** in the lower-left corner.

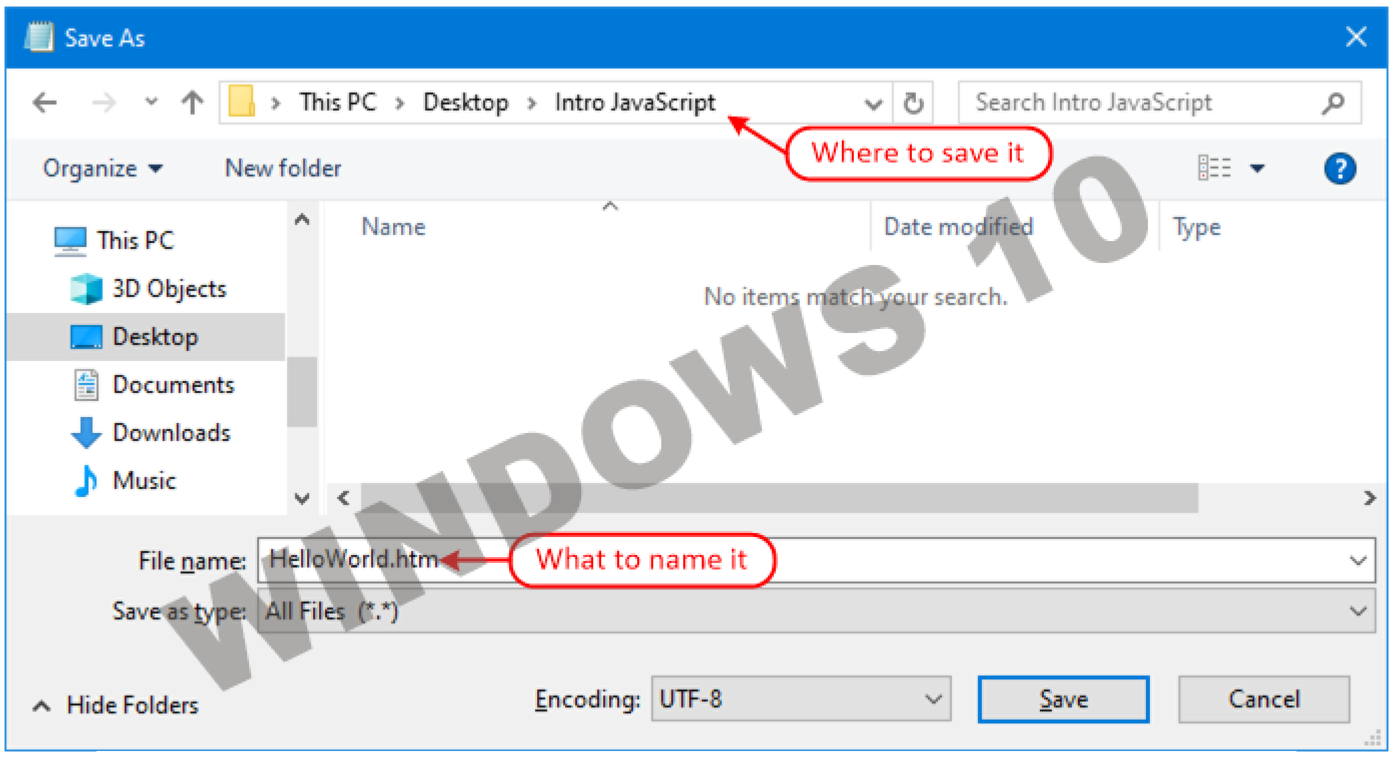
2. Type or copy and paste the following tags needed for an HTML5 webpage:

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body>
</body>
</html>
```

3. To save the page, choose **File** > **Save** (or press **CTRL + S** (Windows) or **COMMAND + S** (Mac)).

4. You'll be asked to provide a **Location.** For pages in this course, use your **Intro JavaScript folder** as the location.

5. Page names will vary, but for this first sample page type *HelloWorld.htm* in for the **Filename**.



Save As

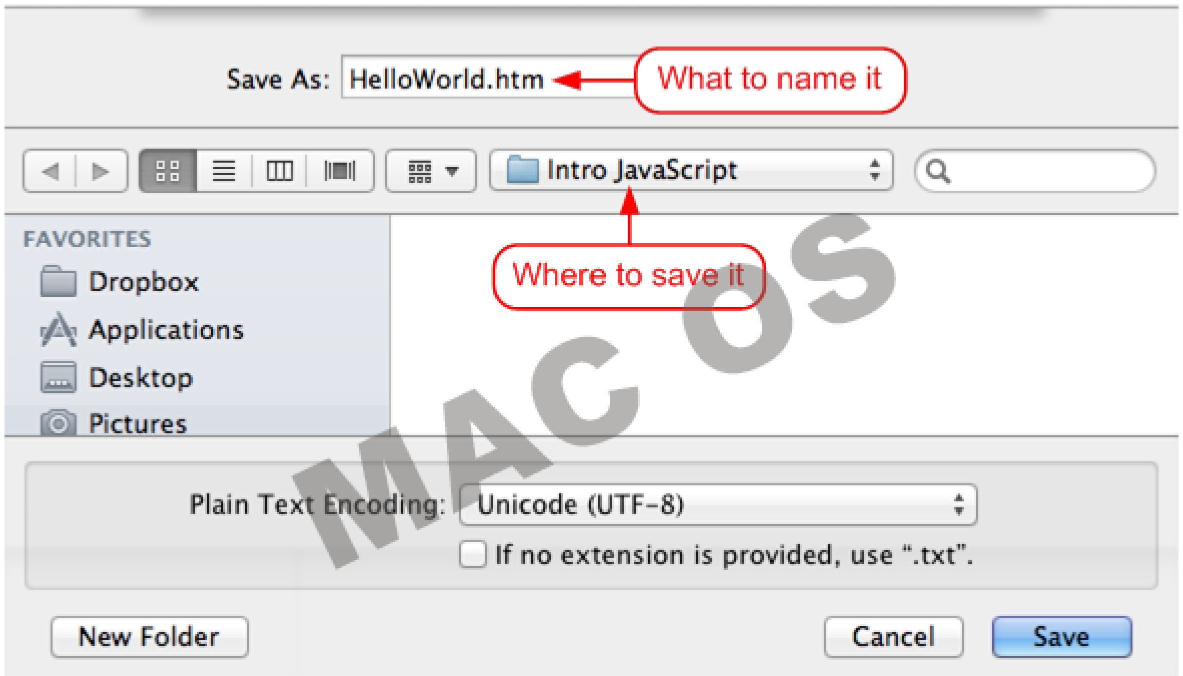
This PC > Desktop > Intro JavaScript

No items match your search.

File name: HelloWorld.htm

Save as type: All Files (*.*)

Encoding: UTF-8



Save As: HelloWorld.htm

Intro JavaScript

Plain Text Encoding: Unicode (UTF-8)

□ If no extension is provided, use ".txt".

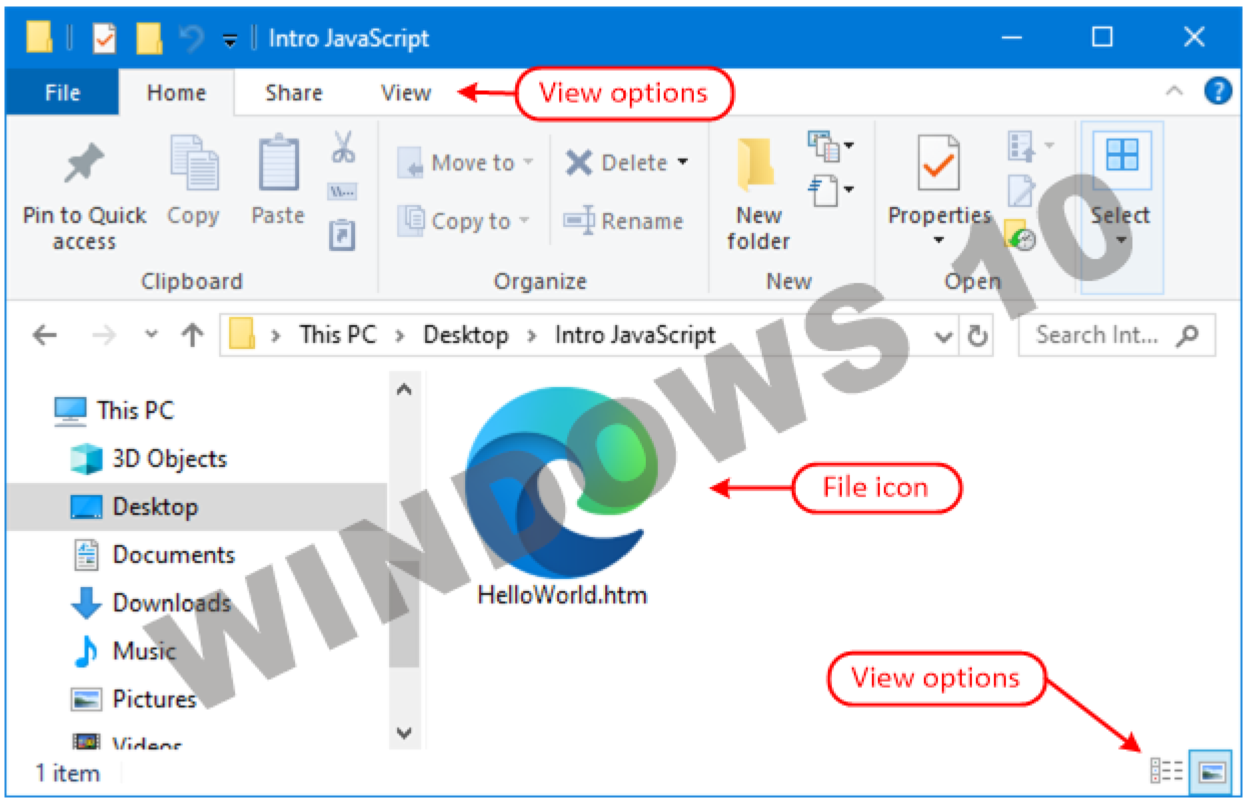
View the Contents of a Folder

We'll be using multiple files in this course, so make sure you understand how to view the contents of a folder without relying on any one program's Open or Save dialog box.

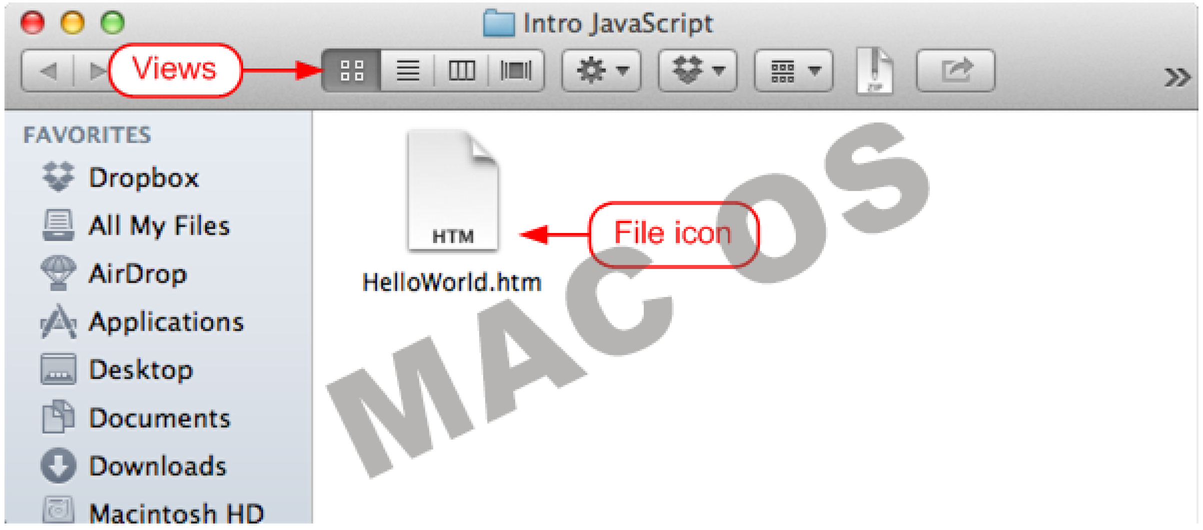
Files and Icons

If you put the *Intro JavaScript* folder on your desktop, you can just double-click its icon there to open it. Or you can use Finder in Mac OS or Windows Explorer in Windows to open the folder. You should see icons for whatever files you've saved so far.

The size and appearance of the icons depend on View settings within the folder. In Windows, the icon usually shows the icon of your default browser.



In Mac OS, you'll see a standard document icon. You can use View options inside the folder to control the size of the icon.



Opening a Webpage File

As with any programming language, you can open a webpage file that contains JavaScript code in two ways:

1. **In a browser, to see what the webpage will look like to users.** To open a page in the default browser on your system, double-click the page's icon. As it stands, the *HelloWorld.htm* page you just created will just be a blank page in any web browser. That's because we haven't added any content to it, so it contains only the minimal HTML code that forms the basic framework of a page.

2. **In an editor, to see and edit the code.** Of course, if you want to see or edit the code, you can't simply click the icon. The code won't be visible in the browser window. Instead, you'll have to open the file with your editor program, just as you would if you were editing the HTML.

An easy way to control whether a page opens in a browser (to preview your page) or an editor (to see and edit code) is is to use the **Open With** function.

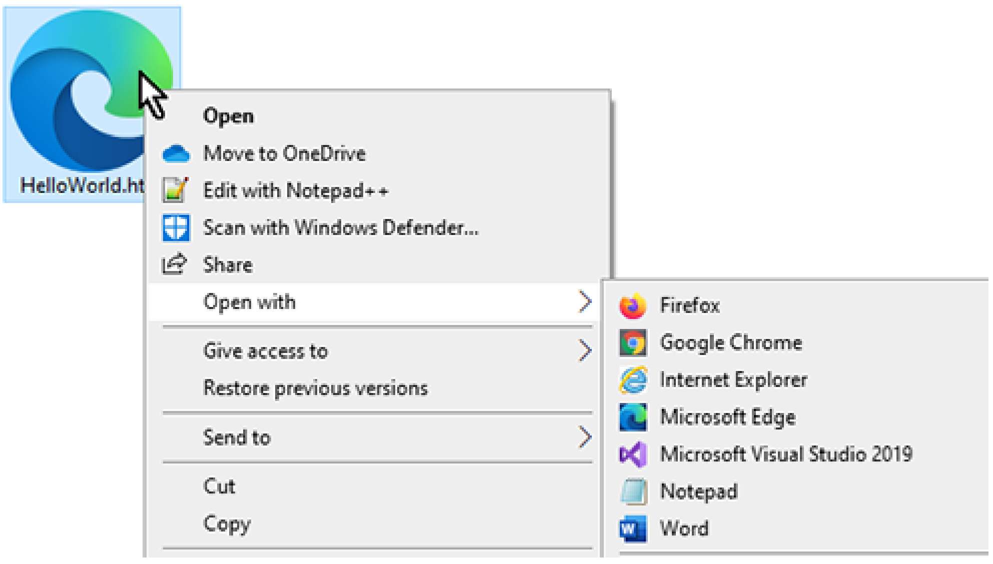
Here are the Steps

1. Right-click (or press **CTRL + click**) the file icon, and choose **Open With**.

2. From here, you can select the specific program you want to use to open the file:

   a. You can opt to open the page in your preferred editor.

   b. You can also choose to view the page in any browsers that you may have installed on your computer.



Checking Your Work

Let's say you've opened your page within the editor and you want to see the effects of your code changes. You could save the file, close it, and then open it in the browser—but there's a simpler way. You can simply keep the file open in both the editor and the browser. Then, all you need to do is save your changes in the page editor you are using and **Reload** or **Refresh** the page in your browser.

For Windows

For Mac

Basically, any time you make a change in the editor you can click the browser to see the effects of the changes. Just make sure you do it in the right order:

STEP 1: Make your change in the editor.

STEP 2: Save the change in the editor.

STEP 3: Reload or Refresh the page in the browser.

As you can see, everything is the same as working with pages that contain CSS and HTML code (without any JavaScript). The only difference is that now you'll be adding in a third language to add more capabilities to your pages beyond what you can do with CSS and HTML alone.

Tip

TIP

When you have multiple program windows open on your desktop, you can hold down the **ALT** key (in Windows) or the **COMMAND** key (in Mac OS) and press **TAB** repeatedly to cycle through the open windows.

So, now you're all set up to get started. You've created your course folder and a file we can work with. Let's start writing JavaScript code!

Chapter 3: Writing JavaScript Code

Adding HTML Tags

First, we're going to look at the HTML tags you need to add to your code to indicate you're using JavaScript.

First and foremost, since JavaScript goes in webpages, you need to start by creating or opening a webpage. We created a page (*HelloWorld.html*) earlier, so now all you have to do is open it in your editor. It's no different from if you were going to add content or CSS or HTML code.

Right now, your page consists of HTML tags that lay out the page. You'll be placing your JavaScript code within that HTML code, so you have to tell the browser where the JavaScript code starts and where it ends. In HTML5, you can use the following tags to do that:

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

The <script> and </script> Tags

As you may have guessed, *script* stems from the fact that JavaScript is a scripting language. The <script> tag tells the browser that what follows is JavaScript code, and the </script> tag tells the browser that this is where the JavaScript code ends. You should put only JavaScript code (no content, CSS, or HTML) between those tags.

Here are the Steps

1. Open HelloWorld.htm in your editor.
2. Add the **<script>** and **</script>** tags under the <body>. You can also leave a blank line or two between the tags to type your code.

```
<!DOCTYPE html>
<html>
<head>
     <title>Page Title</title>
</head>
<body>
<script>
</script>



</body>
</html>
```

3. Don’t forget to Save your changes.

Now, we'll look at how you can add comments to your JavaScript, just as you may have added comments to your HTML code in the past.

Adding Programmer Comments

JavaScript Comments

JavaScript allows you to include programmer comments. These are notes you write in your code, either as reminders to yourself or notes to other programmers when working in teams. You've probably already encountered comments when working with HTML.

There are two ways to write comments in JavaScript:

- **For a single-line comment,** you can start the line with two slashes. Here’s an example of a single-line comment in JavaScript:

```
//This is a single-line JavaScript comment
```

- **For multiline comments,** you can use /* to start the comment and */ to end the comment. (Some of you might recognize those as the same characters used to start and end comments in CSS.) Here's an example of a multiline comment in JavaScript:

```
/* This is a multiline JavaScript comment
You don't have to mark each line as a comment
You just have to indicate where all the comment lines end */
```

You can also use multiline comment marks for single-line comments.

Even though the /* and */ are typically associated with multiline comments in JavaScript, you can use them for single line comments too, like this:

```
/* This is also a single-line JavaScript comment */
```

The JavaScript comments are used only inside the JavaScript code. That means they should be placed between the <script> and </script> tags.

HTML Comments

You may want to add comments outside of your JavaScript tags. For this you use regular HTML comments, which start with <!-- and end with -->. If you wanted to add HTML comments to introduce the purpose of the <script> and </script> HTML tags, they might look something like this:

```
<!-- The script tag marks start of JavaScript code -->
<!-- The /script tag marks end of JavaScript code -->
```

Comments don't actually affect how the code runs, so let’s go ahead and add some HTML and JavaScript comments to your code now.

Here are the Steps

1. Open HelloWorld.htm in your editor.

2. Add the following HTML comment between the <body> and <script> tags.

```
<!-- The script tag marks start of JavaScript code -->
```

3. Next add the following HTML comment between the <script> </script> and <body> </body> tags.

```
<!-- The </script> tag marks end of JavaScript code -->
```

4. Now add the following JavaScript comment between the <script> and </script> tags.

```
//Inside the script tags you type JavaScript code and JavaScript comments
```

5. Now click **Save** so you have the HTML and JavaScript placeholders.

Sneak Peek

Your HelloWorld.htm code should now look something like this:

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>
<body>
  <!-- The script tag marks start of JavaScript code -->
  <script>
  //Inside the script tags you type JavaScript code and JavaScript comments
</script>
  <!-- The /script tag marks end of JavaScript code -->
</body>
</html>
```

Okay, so now you've got your HTML tags set up and you've added some comments. You're ready to add the actual JavaScript code. We'll do that next.

Adding JavaScript

To dip our toes into JavaScript, we're going to use a favorite trivial example of most programmers. You'll be programming your Webpage so that when the page first opens, a message pops up in a small window that reads "Hello World." With this example, you'll get to try out the basic syntax of JavaScript methods.

JavaScript Methods

A *method* is a part of the JavaScript language that performs a specific operation. You'll see many methods throughout this course. JavaScript methods generally use this syntax:

```
method(parameters...)
```

Here, the *method* is usually a word that briefly describes what the method does, and parameters refers to one or more pieces of information that get passed to the method.

Example: Alert Method

We're going to use the method *alert()* to display the message. This method uses this syntax:

```
alert("Text to Display")
```

As you can see, in this example the method is indicated by the word *alert* and the parameters are indicated by the phrase “Text to Display” (which follows in the parentheses). The parameter here specifies the text that you want the alert to show. You can type in any text you want for the parameter.

Basically, the alert method tells the browser that you want to pop a message out in a box, and the text you put as the parameter is the content of that message.

Important

JavaScript is case-sensitive, which means the keywords that define methods and other language features must be typed using specific uppercase and lowercase letters. You'll learn all about these rules, as well as resources for quick, as-needed assistance, throughout this course. But for now, it means that for the *alert* method to work, the word alert must be typed in all lowercase letters.

It's not the same for the parameter ("Text to Display") in this instance. The parameter is *content*, not code. You don't have to worry about the browser being able to "read" the content; content is for people to read. The browser only needs to read the code. So you can use uppercase, lowercase, or any combination for your display text.

So let's add this code to our page.

Here are the Steps

1. Since we want an alert box, add the following code between your JavaScript comment and </script>:

```
alert("Text to Display")
```

2. We want our alert box to display the words *Hello World*, so go ahead and replace the phrase *"Text to Display"* with *"Hello World"*. So, your alert script should match the following code:

```
alert("Hello World")
```

3. Be sure to **Save** the changes.

Sneak Peek

Your HelloWorld.htm code should now look something like this:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <!-- The script tag marks start of JavaScript code -->
  <script>
  //Inside the script tags you type JavaScript code and JavaScript comments
  alert("Hello World")
</script>
  <!-- The /script tag marks end of JavaScript code -->
</body>
</html>
```

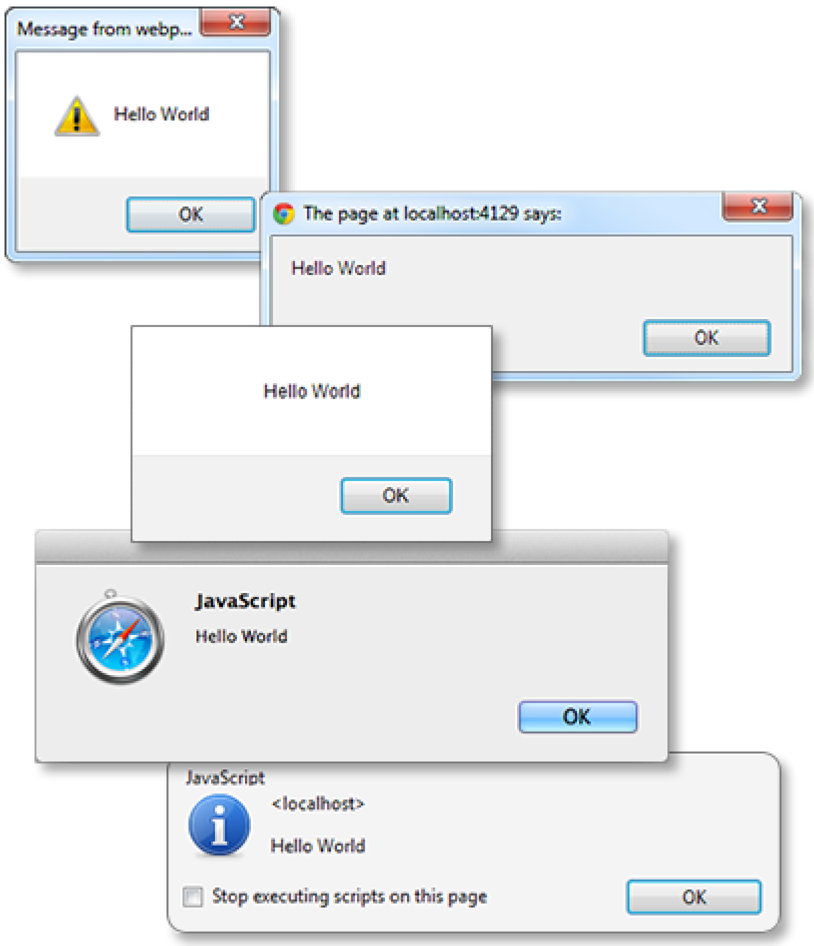
Viewing Your Page in Your Browser

Like CSS and HTML code, JavaScript code is executed in the web browser. So, to test it out, you have to:

1. Save the page.

2. Open the page in a browser.

When you do, the browser will execute the JavaScript code and display an alert box with the words *Hello World*. The exact appearance of that box depends on the brand and version of your web browser.



Hello World

The page at localhost:4129 says:
OK

Hello World

Hello World

Hello World

JavaScript
<localhost>
□ Stop executing scripts on this page
OK

Hello World

Note

If the page is already open in a browser, simply reload or refresh the page.

Did it work? If so, then click the **OK** button to close the alert. That's really all the *alert ()* method does. You've now written your first JavaScript code and executed it.

If yours didn't work, there's probably just a typographical error in your code somewhere. Here are some of the common issues:

- Make sure you type the word *alert* in all lowercase. Remember, that part is case-sensitive.
- Make sure the phrase *Hello World* has both parentheses and quotation marks around it.

Take a close look at your code, compare it to the code provided in the lesson, and try again. If you are really having trouble, you can copy and paste that code into your own *HelloWorld.htm* file.

Now, let's Review and summarize what you've learned here..

Review

This lesson was mostly about getting geared up to learn JavaScript. To summarize the key points:

- **What is JavaScript?:** You learned that JavaScript is a scripting language used in webpages along with CSS and HTML. JavaScript has been around for many years and is supported by all the major Web browsers. You also discovered that JavaScript is not the same as Java. Java is a separate compiled language that's never typed directly into webpages.

- **Are You Ready to Work with JavaScript?:** We looked at the importance of keeping your work organized. You saw that the techniques for creating, editing, and viewing webpages are exactly the same as the techniques for webpages that don't contain any JavaScript.

- **Writing JavaScript Code:** You now know that the <script> tag is used to tell the user agent (web browser) where the JavaScript code starts and that the </script> tag shows where the JavaScript code ends. You also learned that single-line comments in JavaScript, start with //, but multi-line comments must start with /* and end with */.

In the next lesson, you'll build on what you've learned in this lesson to start controlling exactly when your JavaScript code executes, such as when the user clicks a link or buttons or right-clicks some item on the page. Controlling exactly *when* your code executes is as important as controlling *what* it does when it executes. So you'll learn some very important skills indeed. See you there!

Next Steps

To finish the lesson, you'll need to complete the steps outlined below. Simply click "Next Up" at the bottom of the page to access the next activity. Or, if you wish to skip around, click the Book Icon in the top-right corner. There you'll find links to all the activities in this lesson. Here are your remaining activities:

- Browse the Resources for Further Learning section. Here you'll find links to helpful online resources relating to the lesson.

- Do the Assignment. Get some hands-on practice applying what you've just learned.

- Take the Quiz. Reinforce what you learned with a short assessment.

- Participate in the Discussion Area. Ask questions about anything that came up in the lesson, and share your insights. This is where we'll create a learning community.

- Be sure to look at the additional resources. When it comes to a topic like this, there's always more to learn. You can find the link to these resources by clicking on the Resources link on the Learning Path.

Lesson 1 Assignment

Goodbye World

For your first assignment, you're going to continue with the trivial exercise to get a little more hands-on practice working with JavaScript Code. We're keeping it simple so that you can focus on the mechanics of it all without worrying about doing something complex. All you're going to do is change the contents of the alert box to read *Goodbye World* rather than *Hello World*.

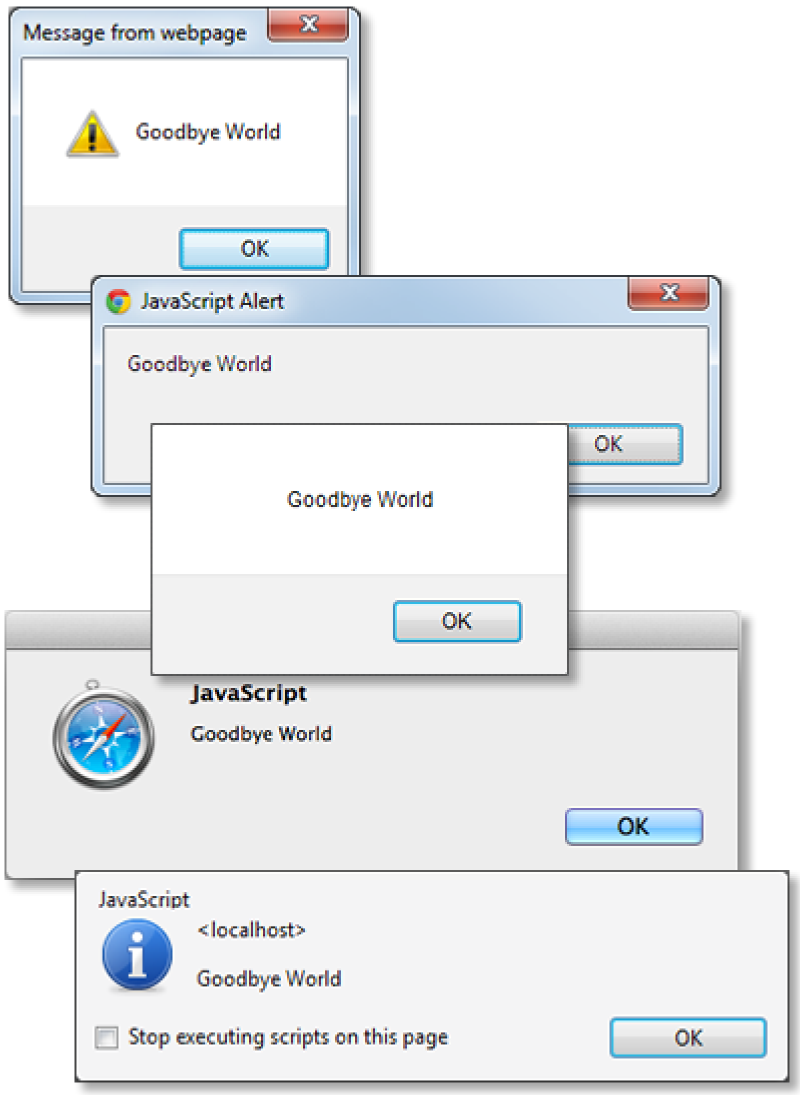
Try it on your own first, without peeking at the instructions. If you can't figure it out, you can click Steps below to see step-by-step instructions.

Make sure you test your code after making the change, and verify that the correct words show up in the alert box when you first open the page in a browser.

- **If the page is already open in a browser,** first close the alert box if it's open (by clicking its OK button). Then click the **Reload** or **Refresh** button in the browser.

- **If the page isn't already open in a browser,** double-click the *HelloWorld.htm* file's **icon** in your *Intro JavaScript* folder.

The page should open in your default browser and show the words *Goodbye World* in an alert box. The new words should appear inside the alert box, the exact appearance of which depends on your browser.



Message from webpage

Goodbye World

JavaScript Alert

Goodbye World

Goodbye World

JavaScript

Goodbye World

JavaScript

<localhost>

Goodbye World

☐ Stop executing scripts on this page

If you were able to pull that off, congratulations! You're well on your way to learning JavaScript. If it was a bit more challenging, you may want to practice the steps a few times, just to get the hang of it. Then we'll see you in Lesson 2!

Lesson 1 Resources for Further Learning

Introduction to JavaScript (https://www.thoughtco.com/what-is-javascript-2037921)

https://www.thoughtco.com/what-is-javascript-2037921

This page offers a brief overview of the JavaScript language and links to related JavaScript websites.

ECMA International (http://www.ecma-international.org/)

http://www.ecma-international.org/

This is the home page for ECMA International, the organization tasked with standardizing ECMAScript, the language that we commonly refer to as JavaScript.

ECMAScript (https://en.wikipedia.org/wiki/ECMAScript)

https://en.wikipedia.org/wiki/ECMAScript

Here is a brief history of ECMAScript, which evolved from JavaScript and is, in fact, the standard for modern JavaScript.

Scripting Languages (https://en.wikipedia.org/wiki/Scripting_language)

https://en.wikipedia.org/wiki/Scripting_language

On this page, you'll find Wikipedia's encyclopedic explanation of scripting languages.

The Script Tag (https://css-tricks.com/the-script-tag/)

https://css-tricks.com/the-script-tag/

This page has a nice description of the HTML <script> tag.

JavaScript Tutorial (https://www.w3schools.com/js/default.asp)

https://www.w3schools.com/js/default.asp

Though brief and blanketed with advertisements, this JavaScript reference and tutorial can be a valuable adjunct to use in conjunction with this course for learning JavaScript.

JavaScript Comments (https://www.w3schools.com/js/js_comments.asp)

https://www.w3schools.com/js/js_comments.asp

From the site mentioned above, here is a reference page for JavaScript comments. You may want to start adding pages like this one to your browser's Bookmarks or Favorites so that they're easy to return to for future reference.