

# Writing and Running Code

by Devmountain Tutorials



## WHAT'S COVERED

This section will explore how to understand code in examples by discussing:

1. WRITING CODE WITH TEXT EDITORS
2. HOW HTML, CSS, AND JAVASCRIPT RUN IN THE BROWSER
3. JAVASCRIPT IN THE BROWSER
4. HOW TO GET STARTED WITH JAVASCRIPT, HTML, AND CSS

## 1. WRITING CODE WITH TEXT EDITORS

To create a website with HTML, CSS, and JavaScript, you'll need a text editor. These are different from word processors like Microsoft Word or Google Documents which allow you to add formatting to your text, like changing fonts or making certain words bold and underlined. Text editors are used to write unformatted text files.

All operating systems come with a basic text editor — Windows comes with Notepad and MacOS comes with TextEdit. These editors will work just fine, but most web developers use third-party editors with specialized tools and features for writing code. Some of the most popular text editors include [Visual Studio Code](#), [Brackets](#), and [Sublime Text](#).

There are also web application-based text editors that allow you to write code directly from within your browser. They're becoming popular for enabling developers to edit code on any device and for automatically saving code to the cloud. These text editors are also great for novice developers since they don't require any setup besides registering for a user account. Popular online text editors include [repl.it](#) and [jsfiddle.net](#).

There can be many subtle differences between different text editors so we recommend putting some thought into deciding which to use. Most editors, including the ones we mentioned, will include features that are essential for writing code like syntax highlighting and code completion.

**Syntax highlighting** is a feature that displays source code in different colors and fonts that correspond with a coding language's grammar. It helps make code easier to read and write because there's a visual distinction between different parts of the code.

```
<html>
  <head>
    <title>Homepage</title>
  </head>
  <body>
    <h1>Hello!</h1>
    
  </body>
</html>
```

*HTML without syntax highlighting*

```
<html>
  <head>
    <title>Homepage</title>
  </head>
  <body>
    <h1>Hello!</h1>
    
  </body>
</html>
```

*HTML with syntax highlighting*

**Code completion** helps reduce the number of keystrokes used to write code. It's similar to how auto-complete or predictive text might help you compose text messages on a mobile device.

```
<html>
  <head>
    <title>Homepage</title>
    link
```

```
  </head>
  <body>
    <h1>
    <img>
  </body>
</html>
```

link

link:atom Emmet Abbreviation

link:css

link:favicon

link:im

link:import

link:manifest

link:mf

link:print

link:rss

link:touch

*HTML editor with code completion*

You may have also seen the term, **IDE**, or **Integrated Development Environment**. These are programs that combine several different tools into a single application. Most IDEs consist of a text editor, tools for building executables, and a debugger. They're often created specifically for writing code in a particular language. For example, **Eclipse IDE** is primarily used to write Java and **PyCharm** is used to write Python.

In general, web developers don't need an IDE to help them run or debug code because most of the code they write will run in a web browser.



#### TERM TO KNOW

#### Syntax highlighting

A feature that displays source code in different colors and fonts that correspond with a coding language's grammar.

#### Code completion

Similar to how auto-complete or predictive text might help you compose text messages on a mobile device, but this helps to reduce the number of keystrokes used to write code.

#### Integrated Development Environment (IDE)

Programs that combine several different tools into a single application.

## 2. HOW HTML, CSS, AND JAVASCRIPT RUN IN THE BROWSER

Whenever you load up a new web page in a browser, it needs to run the code that makes up the web page

(the HTML, CSS, and JavaScript) in order to make the contents of the page appear on the screen. You can think of it like a factory that takes in raw materials — the code — and outputs a product — the web page.

In general, code is loaded and executed in the order it appears on a page, from top to bottom. Every time you navigate to a new page, the process will start over again. The browser will wipe away any code related to the previous page and load in code for the new page. Then, it will run the code and make the contents of the new page appear on the screen. Also, each browser tab has its own “world” and runs code separately from other tabs. This ensures that code in one tab won’t conflict with code in another tab. It’s also a good security measure, because it prevents potentially sensitive information from one tab from leaking to another tab.

---

## 3. JAVASCRIPT IN THE BROWSER

JavaScript is a lot more complex than HTML and CSS, so it warrants an explanation that’s a bit more detailed.

One way to run JavaScript in the browser is by embedding it into your HTML. Then, whenever the browser encounters a block of JavaScript code, it will execute that code as JavaScript (instead of HTML) in the order that each line of code appears.

Note that you don’t have to **compile** or transform the code into a different form so that the browser can run it. Instead, JavaScript is **interpreted** — the code you see is processed and executed directly by the browser. To be more accurate, modern browsers mostly use a strategy called **just-in-time compiling** to improve performance. Browsers will compile JavaScript source code into a format that can run faster. However, you never have to worry about compiling JavaScript yourself, so it’s still considered an interpreted language and not a compiled one.

Another way to put JavaScript into a web page is by importing it from an external file like so:

```
<script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>
```

When the browser sees this, it will load and execute the contents of the file at that URL.



### BIG IDEA

By the way, the URL inside the quotation marks in the script tag above is a valid URL. You can view the contents of the document at that URL by entering it into your browser like any other URL. The code inside isn’t very readable. Still, you might find it interesting!



### TERM TO KNOW

#### Compile

Transform the code into a different form so that the browser can run it.

#### Interpret

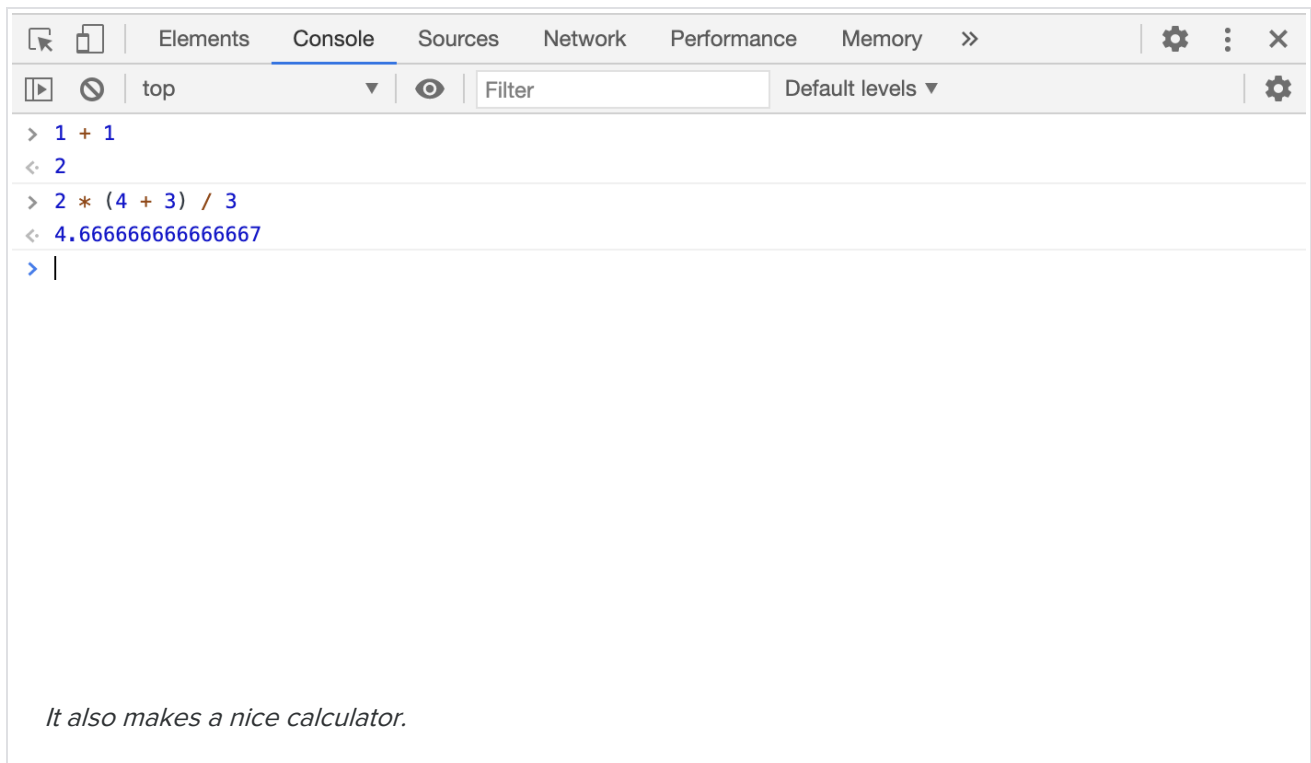
The code you see is processed and executed directly by the browser.

#### Just-in-time compiling

Modern browsers will compile JavaScript source code into a format that can run faster and improve performance.

## 4. HOW TO GET STARTED WITH JAVASCRIPT, HTML, AND CSS

The best way to experiment with JavaScript is through your browser's JavaScript console. Most browsers come with a powerful suite of web developer tools (there are many articles online that will walk you through how to do this, [like this one](#)), so there's a lot less setup involved and zero additional software to install. The JavaScript console allows you to execute lines of JavaScript and view their output, interactively, so it's a great way to learn.



Another way to practice writing HTML, CSS, and JavaScript is through platforms like [codepen.io](#). Codepen is nice because it's geared specifically for front end web developers. In that sense, it's more of a code playground rather than a text editor because you can only use it for HTML, CSS, and JavaScript so it can be slightly more beginner-friendly than online code editors like [repl.it](#). The [trending page on Codepen](#) is a great look at the cool experiments and projects web developers have created with just HTML, CSS, and JavaScript.

If you like data science and data visualizations, you might be interested in [alpha.iodide.io](#) or [observablehq.com](#). They're not as beginner-friendly as Codepen since their focus is on creating data visualizations with JavaScript, but it's interesting to see JavaScript being used for something besides creating software!



### TERMS TO KNOW

#### Code completion

Similar to how auto-complete or predictive text might help you compose text messages on a mobile device, but this helps to reduce the number of keystrokes used to write code.

**Compile**

Transform the code into a different form so that the browser can run it.

**Integrated Development Environment (IDE)**

Programs that combine several different tools into a single application.

**Interpret**

The code you see is processed and executed directly by the browser.

**Just-in-time compiling**

Modern browsers will compile JavaScript source code into a format that can run faster and improve performance.

**Syntax highlighting**

A feature that displays source code in different colors and fonts that correspond with a coding language's grammar.