

## JavaScript Reference Guide: A Beginner's Introduction to Pure JavaScript



JavaScript is one of the most widely used programming languages, and its simplicity makes it appealing to beginners. Download this free guide of essential commands if you are looking to program in JavaScript.

By: Christopher Tozzi

There's a lot to like about [JavaScript](#). As [programming languages](#) go, it's pretty simple and easy to learn. Its straightforward syntax and predictable paradigms make it a great language for beginners.

JavaScript is also one of the most widely used languages today. That means it's easy to find tutorials and documentation related to JavaScript coding.

Combine these advantages with the fact that JavaScript has been the go-to solution for building dynamic and interactive web pages for decades, and it's easy to see why JavaScript code runs virtually everywhere today.

To help aspiring JavaScript coders get started with this flexible and dynamic language, we've developed a reference guide that covers the fundamentals of JavaScript programming. Below, you'll find simple examples of how to perform core tasks using JavaScript — such as working with strings and variables, doing math, managing arrays, and creating functions. There's certainly much more to JavaScript than what you'll find here, but this sample code will help get you started in learning this ubiquitous programming language.

## Two Caveats

Before diving in, please heed a couple of caveats. First, because we want to stick to the basics, this reference guide focuses on "pure" JavaScript code and avoids referencing code that requires use of a particular JavaScript framework. If you're developing JavaScript in the real world, you may benefit from using a framework or library, such as Angular or React, which can make it simpler to implement complex JavaScript code. But it's best to start by learning pure JavaScript.

Second, we're sticking here to coding examples that you'd find in client-side JavaScript code. In other words, this reference guide shows you how to use JavaScript to help develop websites and web apps. It's also possible to use JavaScript as a server-side programming language via [Node.js](#); in fact, JavaScript's ability to support both client-side and server-side use cases is part of what makes it so powerful. But server-side programming is beyond the scope of this introductory reference guide. Learn the basics of JavaScript first, and save Node.js for another day.

Action	Command
<b>Strings</b>	
Define a string	let string = "string value";
Concatenate strings	const string1 = "Hello"; const string2 = "World!"; const combined = string1 + " " + string2;
<b>Variables</b>	
Set a variable as integer	let x = 1;
Increment a variable	let x = 1; const y = x++;
<b>Displaying data</b>	
Print a string to the console	console.log(string);
<b>Math</b>	
Add numbers	x = 1; y = 2; z = x + y;
Subtract numbers	x = 1; y = 2; z = y - x;
Multiply numbers	x = 2; y = 2; z = y * x;
Divide numbers	x = 2; y = 2; z = y / x;
<b>Arrays</b>	
Create an array	const my_array = ["value1", "value2", "value3"];

Add item to array	<pre>const my_array = ["value1", "value2", "value3"]; my_array.push("value4");</pre>
Remove item from array	<pre>const my_array = ["value1", "value2", "value3"]; delete my_array[2];</pre> <p>Note: The delete operator removes the object at the selected index but does not change the length of the array, so an undefined index value takes its place.</p>
Sort an array	<pre>const my_array = ["value1", "value2", "value3"]; my_array.sort();</pre>
<b>Dates</b>	
Select day of week	<pre>var d = new Date(); a = d.getDay();</pre>
Select month of year	<pre>var m = new Date(); a = m.getMonth();</pre>
<b>Comments</b>	
Insert a single-line comment	<pre>// This is a comment let x = 1; // This is also a comment</pre>
Insert a multi-line comment	<pre>/* Comment text begins on this line And continues on this line And ends on this line */</pre>
<b>Files</b>	
Read a local file	<pre>xmlhttp.open("GET","some_file.txt",true);</pre>
Note: Writing files to disk is typically not possible using client-side JavaScript	

<b>Searches and regexes</b>	
Search for regex in string	<pre>const string = "1 2 3 x y z"; const regex = /^[^w]*/g;</pre>
Search for value in array	<pre>my_array.includes(value);</pre>
<b>Conditionals</b>	
Create a for loop	<pre>for (let n = 0; n &lt; 10; n++) { console.log(n); }</pre>
Create a while loop	<pre>let n = 0; while (n &lt; 10) { console.log(n) n++; }</pre>
Create if and else if statements	<pre>let n = 1; if (n &lt; 0) { console.log(n , 'is negative'); } else if (n &gt; 0) { console.log(n , 'is positive'); }</pre>
<b>Functions</b>	
Define a function	<pre>function my_function() { let x = 1; }</pre>
Call a function	<pre>my_function();</pre>