

Lesson

Weekend

Introduction to Programming

(/introduction-to-programming)

/ JavaScript and Web Browsers

(/introduction-to-programming/javascript-and-web-browsers)

/ Arithmetic

Text

Cheat sheet

Note: If you are a remote student and you are going through these lessons with a pair, you'll need to screen share your Chrome DevTools console with your pair (more instructions below). We prioritize using the browser console (and other developer tools) because it's such a vital tool to be proficient with as a developer.

Arithmetic, Expressions, and Operators

Remember, web browsers can only do three things: display content with HTML, style it with CSS, and change what's displayed with JavaScript. Because all browsers support JavaScript, it's the de facto programming language of the web.

We're going to take a detour away from the aesthetically pleasing and obviously useful world of web pages so that we can learn some programming fundamentals. But don't worry! We'll be back in no time.

Open Your DevTools Console

We'll start by writing JavaScript in Chrome's DevTools console. To access it:

- Click on the three dots at the upper right corner of the browser screen.
- Go to the *More Tools* menu.
- Click *Developer Tools*. This will open the developer tools side bar. This "side bar" could also be at the bottom of your browser window.
- Next, select the *Console* tab within the developer tools.

Here are the shortcuts to access the DevTools:

- Windows:
 - Ctrl + Shift + j
 - Ctrl + Shift + i
- Mac:
 - Cmd + Option + i
 - F12

Writing a JavaScript Expression

The angle bracket `>` represents the DevTools console prompt. At the `>` prompt, you can type JavaScript and see what it does. Try typing `1 + 2;` and pressing Enter.

Congratulations! You just ran your first JavaScript code.

You should see something like:

```
> 1 + 2;  
3
```

JavaScript took your code `1 + 2`, evaluated it, and returned the result: `3`.

This is called an **expression**. An expression in computer programming is code that evaluates a value.

Note the semicolon at the end of the line. Semicolons are a bit of a tricky subject in JavaScript. When we execute our code, JavaScript interprets it on the fly into code that our machines can read. In the process, it automatically adds semicolons between sections of our code. *However*, there are certain situations where JavaScript incorrectly adds a semicolon — and these situations are a bit obscure for beginners. Some developers add semicolons themselves to be thorough while others only add them when needed because JavaScript will do it automatically. However, you need to know those gotcha situations to be in the latter camp. **For that reason, we will expect you to consistently use semicolons for JavaScript as we describe here and in future lessons.**

Now enter these into the DevTools console:

- `4 - 3;`
- `5 * 6;`
- `10 / 2;`
- `9 / 2;`
- `7 + 8 * 9;`
- `(7 + 8) * 9;`

These are all **expressions** as well because they evaluate to a value. Note that they all have semicolons at the end of the line. Even though these expressions will work fine in the console without semicolons, we expect you to use semicolons when writing JavaScript.

JavaScript Operators

`+`, `-`, `*`, and `/` are called **operators**. An operator is a special character (or characters) that indicates an action to be performed.

NaN

Try dividing 0 by 0. The result, `NaN`, stands for *not a number*. **In JavaScript, `NaN` is actually considered a type of number (bizarre as that may seem).**

Infinity

Try dividing any other number by 0. The result, `Infinity`, **is also a number in JavaScript.**

[Previous \(/introduction-to-programming/javascript-and-web-browsers/javascript-data-types\)](#)

[Next \(/introduction-to-programming/javascript-and-web-browsers/practice-arithmetic\)](#)

Lesson 7 of 75

Last updated March 24, 2023

[disable dark mode](#)



Epicodus (<http://www.epicodus.com>)

© 2023 Epicodus (<http://www.epicodus.com/>), Inc.