**01 – Fundamentals Part 1**

Introduction

Let’s dive right into JavaScript!

Lesson Overview

This section contains a general overview of topics that you will learn in this lesson.

    How do you declare a variable?

    What are three different ways to declare a variable?

    Which one should you use when?

    What are the rules for naming variables?

    What are operators, operands, and operations?

    What is concatenation and what happens when you add numbers and strings together?

    What are the different types of operators in JavaScript?

    What is the difference between == and ===?

    What are operator precedence values?

    What are the increment/decrement operators?

    What is the difference between prefixing and postfixing them?

    What are assignment operators?

    What is the Unary Plus Operator?

How to Run JavaScript Code

All JavaScript we will be writing in the majority of the Foundations course will be run via the browser. Later lessons in

Foundations and the NodeJS path will show you how to run JavaScript outside of the browser environment. Outside of these lessons,

for now you should always default to running your JavaScript in the browser unless otherwise specified, otherwise you may run

into unexpected errors.

The simplest way to get started is to simply create an HTML file with the JavaScript code inside of it. Type the basic HTML

skeleton into a file on your computer somewhere:

<!DOCTYPE *html*>

<html>

<head>

  <meta *charset*="UTF-8">

  <title>Page Title</title>

</head>

<body>

  <script>

    // Your JavaScript goes here!

    console.log("Hello, World!")

  </script>

</body>

</html>

Save and open this file up in a web browser (you can use “Live Server” on Visual Studio Code to do this!) and then open up the browser’s console by right-clicking on the blank webpage and selecting “Inspect” or “Inspect Element”. In the ‘Developer Tools’ pane find and select the ‘Console’ tab, where you should see the output of our console.log statement.

console.log() is the command to print something to the developer console in your browser. You can use this to print the results from any of the following articles and exercises to the console. We encourage you to code along with all of the examples in this and future lessons.

Another way to include JavaScript in a webpage is through an external script. This is very similar to linking external CSS docs to your website.

  <script *src*="javascript.js"></script>

JavaScript files have the extension .js similar to .css for stylesheets. External JavaScript files are used for more complex scripts.

Variables

You can think of variables as simply “storage containers” for data in your code.

Until recently there was only one way to create a variable in JavaScript — the var statement. But in the newest JavaScript versions we have two more ways — let and const.

      1. This variable tutorial (https://javascript.info/variables) will explain everything you need to know! Be sure to do the Tasks at the end. Information won’t stick without practice!

The above tutorial mentioned this, but it’s important enough to note again: let and const are both relatively new ways to declare variables in JavaScript. In many tutorials (and code) across the internet you’re likely to encounter var statements. Don’t let it bother you! There’s nothing inherently wrong with var, and in most cases var and let behave the same way. But sometimes the behavior of var is not what you would expect. Just stick to let (and const) for now.

Numbers

Numbers are the building blocks of programming logic! In fact, it’s hard to think of any useful programming task that doesn’t involve at least a little basic math… so knowing how numbers work is obviously quite important. Luckily, it’s also fairly straightforward.

      1. This W3Schools lesson (https://www.w3schools.com/js/js\_arithmetic.asp) followed by this one (https://www.w3schools.com/js/js\_numbers.asp), are good introductions to what you can accomplish with numbers in JavaScript.

      2. This MDN article (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\_steps/Math) covers the same info from a slightly different point of view, while also teaching you how to apply some basic math in JavaScript. There’s much more that you can do with numbers, but this is all you need at the moment.

      3. Read through (and code along with!) this article (https://javascript.info/operators) about operators in Javascript. Don’t forget to do the “Tasks” at the bottom of the page! It will give you a pretty good idea of what you can accomplish with numbers (among other things!) in JavaScript.

Assignment

Try the following exercises (and don’t forget to use console.log()!):

  1. Add 2 numbers together! (just type console.log(23 + 97) into your html file)

  2. Add a sequence of 6 different numbers together.

  3. Print the solution to the following equation: (4 + 6 + 9) / 77

      Answer should be approximately 0.24675

  4. Let’s use variables!

      Type this statement at the top of the script tag: let a = 10

      In the console console.log(a) should print 10

      Try the following in the console: 9 \* a

      and this: let b = 7 \* a (returns undefined \*) and then console.log(b)

  5. You should be getting the hang of this by now… try this sequence:

      Declare a constant variable max with the value 57

      Set another variable actual to max - 13

      Set another variable percentage to actual / max

      If you type percentage in the console and press enter you should see a value like 0.7719

  6. Take a few minutes to keep playing around with various things in your script tag. Eventually, we will learn how to actually make those numbers and things show up on the webpage, but all of this logic will remain the same, so make sure you’re comfortable with it before moving on.

\* As you might have noticed by running Javascript code in the console, the console prints the result of the code it executes (called a return statement). You will learn more about these in the next lessons, however for now it is good to remember that a declaration with an assignment (such as let b = 7 \* a) returns undefined and so you cannot declare and assign a value to a variable and read its value in the same line.

Additional Resources

This section contains helpful links to other content. It isn’t required, so consider it supplemental.

The precise differences between var and let is explained in javascript.info (https://javascript.info/var).

Knowledge Check

This section contains questions for you to check your understanding of this lesson on your own. If you’re having trouble answering a question, click it and review the material it links to.

      Name the three ways to declare a variable

      Which of the three variable declarations should you avoid and why?

      What rules should you follow when naming variables?

      What happens when you add numbers and strings together?

      How does the Modulo (%), or Remainder, operator work?

      Explain the difference between == and ===.

      When would you receive a NaN result?

      How do you increment and decrement a number?

      Explain the difference between prefixing and postfixing increment/decrement operators.

      What is operator precedence and how is it handled in JS?

      How do you access developer tools and the console?

      How do you log information to the console?

      What does unary plus operator do to string representations of integers? eg. +”10”

**02 – Fundamentals Part 2**

Introduction

There are a few extremely common types of data that you will encounter in JavaScript, and these lessons on fundamentals will give us a really strong foundation in all of them. Before we start digging deep, however, this article (https://javascript.info/types) will give you a quick overview of the most common ones.

Lesson Overview

This section contains a general overview of topics that you will learn in this lesson.

    Name the eight data types in JavaScript.

    Understand the difference between single, double, and backtick quotes.

    Embed a variable/expression in a string.

    Understand what a method is.

    Name the three logical operators.

    Understand what the comparison operators are.

    Understand what conditionals are.

    Understand what nesting is.

    Understand what truthy and falsy values are.

Strings

Depending on what kind of work you’re doing, you might end up working more with pieces of text rather than numbers. A string is simply a piece of text… and is a fundamental building block of the language.

    1. Read and code along with yet another MDN tutorial (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\_steps/Strings) on the topic.

    2. Go through this lesson (https://www.w3schools.com/js/js\_string\_methods.asp) to learn a bit more about what you can do with strings… be sure to take a peek at the String Reference (https://www.w3schools.com/jsref/jsref\_obj\_string.asp) page near the bottom, and do the exercises at the end!

    3. Vocabulary time: a method is a bit of functionality that is built into the language or into specific data types. In the previous W3Schools exercise (https://www.w3schools.com/js/js\_string\_methods.asp), you learned a few methods that can be used on strings, such as replace and slice. An exhaustive list of methods that can be used on strings can be found here (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/String).

Conditionals

Now it’s time for the fun stuff… So far we haven’t done much with our programming that you couldn’t do with simple math skills. Sure, we’ve told our computer how to do the math, so that makes it quicker, but the essence of programming is teaching the computer how to make decisions in order to do more involved things. Conditionals are how we do that.

    1. Step one in learning about conditionals is making sure you have a good grasp on comparisons (http://javascript.info/comparison).

    2. This tutorial (https://www.w3schools.com/js/js\_if\_else.asp) is a great first glance at conditionals in JavaScript.

    3. This tutorial (http://javascript.info/logical-operators) will teach you about logical operators. A little heads up regarding this reading’s tasks; there will be questions where you see alert() (with a number or string inside of the parenthesis), what’s happening here will be discussed later in the curriculum. Some of the answers may not make sense now but they are accurate and will come to be understood as you progress in the curriculum. Don’t worry too much about it now!

    4. This article (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building\_blocks/conditionals) reinforces the concept and provides several interesting examples of how you could use it building websites.

    5. This article (http://javascript.info/ifelse) covers the same basic concept (read through it as a review!) and - more importantly - offers the usual ‘tasks’ at the bottom of the page!

    6. This article (https://javascript.info/switch) teaches you about the switch statement, which comes in handy when you have multiple conditions.

Assignment

To give you a good bit of practice, we have created some replit.com exercises for you to play with. We believe that it’s best to practice programming on your own computer, rather than in an online environment, but we’ll get to that soon enough.

Be sure to do the lessons in order presented here. Pressing “run” at the top will run the code. Read all directions, watch the terminal, and read all the errors. Don’t forget to use ‘console.log’ extensively.

To get started, create a free replit account and click “Fork” or “Remix” in order to have access to the exercises. Note: Feel free to browse the files on the left column in order to gain familiarity with it.

Exercise 1

In this exercise, you will be working out of the file called troubleshooting.js

===== Troubleshooting =====

The function below should log the number 2, however it does not, see if you can fix it!

Be sure to fix it in the spirit of the code, do not hard code the result.

function troubleshooting() {

  const a = 1;

  const b = 1;

  let result;

  result = "a" + "b";

  return result;

}

Exercise 2

You will be working out of script.js, and you will use the console in the ‘webview’ pane to check your work. To access the console, click the wrench icon, which is located on the right side of the address bar within the ‘webview’ pane.

The code below tells the browser to ask you for a number then if that number is `6`, it returns `true` otherwise it returns `false`

Change this code so it returns `true` when the number is greater than or equal to 10, and false if it is less than 10

number = Number(prompt("enter a number"));

function numberChecker() {

  if(number === 6) {

    return true;

  } else {

    return false;

  }

}

Exercise 3

You will be working out of math.js

Lets do some math!

Some rules first:

- Replace the strings to the right of the = with the math expression they describe.

- Do not manually enter the answers to the equations. For example, `const a = 9` would be incorrect as 9 is the answer to the equation you're being asked to write out

const a = "one plus eight"

const b = "22 times three"

const c = "the *\*remainder\** of 5/4"

const d = "the variable 'a' minus 17"

const e = "the sum of the previous four variables"

Exercise 4

You will be working out of follow.js

- After each step, run the code to make sure your code still works!

- If your code fails to run, or you see a "ReferenceError" in the console, review your code and make sure you have completed all of the objectives.

- Don't forget you can put "console.log" anywhere to see what your values are at any time.

  ===== Step 1: =====

  Look at the code below and make a prediction of what the output will be.

  After making your prediction, press 'Run' at the top and look at the output in the console. If you were surprised by anything, go back and look at the code to see what's going on.

const birthYear = 1948;

const thisYear = 1965;

const firstName = "Carlos";

const lastName = "Stevenson";

const greeting = "Hello! My name is " + firstName + " " + lastName + " and I am " + (thisYear - birthYear) + " years old.";

console.log(greeting);

  ===== Step 2: =====

  Once you understand the code snippet above, delete it. Then, using the following instructions, recreate the snippet on your own under "Your code goes here".

  ---------------------------------------------------------------

  1. Create 4 variables: firstName, lastName, thisYear, and birthYear

  2. Create a 5th variable, greeting, that is constructed from the previous 4

  (it should contain a greeting with the person's full name and their age)

  3. Print greeting with console.log

  4. Press 'Run' to ensure your code works

  ---------------------------------------------------------------

  In order to make the tests pass you will need to use these exact values. The wording also needs to be an exact match. If the tests fail, check spacing, capitalization, and punctuation:

  birth year = 1948

  this year = 1965

  first name = Carlos

  last name = Stevenson

  The greeting should say: "Hello! My name is Carlos Stevenson and I am 17 years old."

//===== Your code goes here =================

  ===== Step 3: =====

  Now that you have the code working again, let's go back over it and, using the instructions below, edit it to make it easier to read.

  1. Go to "Testing your code" below

  2. Comment out the lines under "Test Step 2"

  3. Uncomment the lines under "Test Step 3" (Notice the difference between them)

  4. Go back to your code and create 2 new variables: "fullName" and "age"

  Do NOT simply type the full name and age into the new variables. Set them using the pre-existing variables, with the calculations that are currently inside of greeting.

  5. Edit the greeting string to use fullName and age instead of the other 4 variables. (the greeting should then look something like: "Hello, my name is " + fullName)

  6. Press 'Run' to ensure your code still works (output should be unchanged)

  ===== Testing your code =====

  - Do NOT edit this section until told to do so.

  - Make sure one and only one of these test steps are commented out at a time

// Test Step 2:

module.exports = {

  testGroup: "a",

  greeting,

  birthYear,

  thisYear,

  firstName,

  lastName

}

// Test Step 3: (Don't forget to comment out lines under Test Step 2)

// module.exports = {

//  testGroup: "b",

//  greeting,

//  birthYear,

//  thisYear,

//  firstName,

//  lastName,

//  fullName,

//  age

// }

Additional Resources

This section contains helpful links to other content. It isn’t required, so consider it supplemental.

  - Regular expressions, commonly known as regex, are a tool that matches or locates patterns in strings for string validation. Although it shouldn’t be your immediate solution this early on, you can still use this resource to understand how websites know that myemail@com isn’t a valid email address. On top of that, other solutions to filter out strings exist, and regex is considered a slow operation.

  - The Net Ninja’s Regular Expressions Tutorial (https://www.youtube.com/playlist?list=PL4cUxeGkcC9g6m\_6Sld9Q4jzqdqHd2HiD).

  - When to avoid regular expressions (https://softwareengineering.stackexchange.com/questions/113237/when-you-should-not-use-regular-expressions).

Knowledge Check

This section contains questions for you to check your understanding of this lesson on your own. If you’re having trouble answering a question, click it and review the material it links to.

      What are the eight data types in JavaScript?

      Which data type is NOT primitive?

      What is the relationship between null and undefined?

      What is the difference between single, double, and backtick quotes for strings?

      What is the term for joining strings together?

      Which type of quote lets you embed variables/expressions in a string?

      How do you embed variables/expressions in a string?

      How do you use escape characters in a string?

      What is the difference between the slice/substring/substr string methods?

      What are the three logical operators and what do they stand for?

      What are the comparison operators?

      What are truthy and falsy values?

      What are the falsy values in JavaScript?

      What are conditionals?

      What is the syntax for an if/else conditional?

      What is the syntax for a switch statement?

      What is the syntax for a ternary operator?

      What is nesting?