Learn Programming in JavaScript

Overview

Project Title: Learn Programming in JavaScript

Target Age Range: 14-15

Necessary Tools:

* Web Browser (Any latest version of a major browser can be used i.e. Google Chrome, Mozilla Firefox, Internet Explorer, Opera, Safari. Google Chrome is used throughout the course.)
* Text editor (Any text editor can be used i.e. Brackets, Sublime Text, Notepad, Notepad++, Atom etc. Brackets is used throughout the course.)

The Aim of the Project: The aim of this project is to grasp programming fundamentals and to learn JavaScript which is one of the most widespread and high-potential languages.

The teaching material includes an introduction to JavaScript, lessons for JavaScript concepts, exercises for each lesson and solutions for these exercises.

About the Author and Preface:

I am a first year undergraduate Computer Science student at UCL. There are many things about computer science that attract me. The most significant one is that the ease of access to free material and ease of use of these material. A decent computer and a decent internet connection is more than enough to start learning a programming language and start building new projects. If you enjoy programming, you are lucky because you can build so many things only with your computer with minimum needs and minimum cost. Even if you are not going to study a computer related degree, learning how to program is still an important qualification in sense of learning how to take different approaches and learning how to create innovative ideas to solve problems. I hope you enjoy the course and find it useful.

Index:

* Introduction to JavaScript
  + - What is JavaScript?
    - Why JavaScript?
    - History of JavaScript
  + Lessons, Exercises &Solutions
  + Final Project
  + Reference List

Introduction to JavaScript:

What is JavaScript?

JavaScript is a high-level, dynamic and interpreted programming language.

Syntax: Derived from the C programming language

Semantics & Design: Influenced by the Scheme and Self programming languages

Programming Style: Object-Oriented, Imperative, Functional

Last Standard Edition: ECMAScript 6, released in June 2015

Why JavaScript?

Apart from HTML and CSS, JavaScript is one of the essential languages of web development, most of the websites benefit from it and all browsers support it. It is also used in building desktop and mobile applications, in game development and in many other areas.

History of JavaScript

JavaScript was originally developed in 1995 by Brendan Eich, an employee of Netscape Communications Corporation. The first name of the language was Mocha. However, it was officially released under the name LiveScript. After a short while it was renamed JavaScript when it was included in the Netscape browser. Although the name JavaScript sounds like it was developed from Java, they have nothing to do with each other apart from minor similarities in syntax.

Although JavaScript is considered as a front-end language, server-side implementation of JavaScript was aired just after its first appearance, it started becoming popular in mid-2000s. JavaScript gained even more popularity with different implementations of the language such as Node.js, jQuery, Underscore.js, CoffeeScript JSON and AJAX.

Most of the time, JavaScript code is embedded in HTML. The most basic operations can be done with JavaScript are: fading them in and out, moving them, resizing them, animation of page elements, validating Web forms to make sure that they are acceptable before being submitted to the server, transmitting information about the user's reading habits and browsing activities to various websites. Web pages frequently do this for web analytics, ad tracking, personalization or other purposes.

Lessons, Exercises & Solutions

If you want to take most out of this course, try to write your own code before looking at the solutions. It doesn’t really matter if you succeed or not. The important thing is to push yourself to learn more programming. After looking at the solutions play around with the code to make the program to perform different actions. Solutions to exercises can be found in the “Solutions” folder. Good luck!

Lesson 1: Our First JavaScript Program

As a tradition in Computer Science, the first program you write when you are learning a programming language is the HelloWorld! program. To write this program in JavaScript, we need to embed JavaScript code in HTML code. For the most part of this course, we will not care much about html. Only basic HTML elements will be used. JavaScript code needs to be embedded between HTML’s <script> </script> elements so that the browser can interpret the JavaScript code.

Here is the HelloWorld program:

JavaScript can also be included using an external JavaScript file like the following code:

1\_HelloWorld.html

1\_HelloWorldExternal.html

Lesson 2: Comments

Comments are very important for programs. They make it easier for other programmers to understand the code written. At the same time, comments recall to the author what a piece of code does when the author looks at the program sometime in the future. The program simply omits comments and they have absolutely no effect on the program.

Here is how comments are inserted into JavaScript code:

2\_Comments.html

Lesson 3: Variables

Variables exist almost in every program. Variables can be integers, floating numbers, characters, strings and Boolean values. In JavaScript the keyword var is used to define variables. Using variables is the simplest way to store values. Values need to be assigned to variable in order to get stored. The assignment operator is =. Variable names cannot include spaces, special symbols such as +, \* and cannot start with a number. Variable names are case sensitive.

Here is how to define variables and assign values to these variables:

3\_Variables.html

Lesson 4: Input / Output

Getting user input and giving the output back are two main components of a program. The simplest way to print a value to the screen is to use document.write(“Text goes in here.”) function. The easiest way to get user input is to use prompt() function.

Here is an example of input and output:

4\_InputOutput.html

Lesson 5: Mathematical Operators

Mathematical operators are one of the most used concepts in JavaScript as they are in all programming languages. The most used mathematical operators we can put in our program are:

+ (addition)

++ (increment)

-- (decrement)

- (subtraction)

\* (multiplication)

/ (division)

% (modulus)

Here is a program using mathematical operators:

5\_MathematicalOperators.html

Lesson 6: Comparison Operators

The comparison operators in JavaScript are:

== (equal to)

=== (identical to (equal and of same type))

!= (not equal to)

!== (not identical to)

> (greater than)

>= (greater than or equal to)

< (less than)

<= (less than or equal to)

Since comparison operators are mostly used with conditional statements, the example use of comparison statements can be seen in the next chapter with conditional statements, since mostly they are used together.

Lesson 7: Conditional Statements

We may not always want to run each line of the code. We may want the program to execute different parts of the code depending on the conditions. That is where we use conditional statements. The first element of conditional statements is if.

Here is an example use of if statement:

If we want the program to execute some other code if the if statement is not met, we use else statement.

Here is an example of else statement:

If we have to include more than two conditions, we use else if statement.

Here is an example with an else if statement:

The short form of an if statement is a ternary statement.

Here is an example of if and ternary statements together:

Instead of using many else if statements, we can simply use the switch statement.

## Here is an example of a switch statement with corresponding if statement:

## If we do not include break statements all lines gets executed. Default statement gets run only if none of the case statements are met.

## 7\_If.html

## 7\_Else.html

## 7\_Else.html

## 7\_Switch.html

## 7\_Ternary.html

Lesson 8: Logical (Boolean) Operators

Logical operators are another useful concept in JavaScript.

! returns true, if operand is false, and false, if operand is true

&& returns true, if both operands are true

|| returns true, if at least one of the operands are true

Here is an example use of logical operators:

8\_LogicalOperators.html

Lesson 9: Loops

Loops provide programs to execute some parts of the code many times without typing the code multiple times. There are 3 different type of loops: For Loop, While Loop, Do While Loop.

The syntax for the for loop:

for (statement 1; statement 2; statement 3){

code to be executed (loop body)

}

Statement 1 is executed before the loop, statement 2 defines the condition for the loop, statement 3 is executed at the end of each iteration.

To understand the for loop better, look at the following example:

While loops are just like for loops. The difference is, only the condition needs to be defined before the loop body.

Look at this while loop example.

Do while loop is the same as while loop except the first execution of the code is guaranteed.

Here is an example of do while loop:

To manipulate the code, we can use the break and continue statements.

Here is an example containing break and continue:

9\_WhileLoop.html

9\_DoWhileLoop.html

9\_ForLoop.html

9\_BreakContinue.html

Final Project:

Open the website.html file and by using the information on the websites that are provided in the reference list. Enhance the interface and if you can, add form validation as well.

Reference List:

* Devdocs, JavaScript, [Online] Available at: http://devdocs.io/javascript/ Accessed: 7 March 2016
* Mozilla Foundation, JavaScript, [Online] Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript Accessed: 20 March 2016
* Solo Learn, JavaScript, [Online] Available at: http://www.sololearn.com/Course/JavaScript/ Accessed: 1 February 2016
* w3schools, JavaScript, [Online] Available at: http://www.w3schools.com/jsref/default.asp Accessed: 12 March 2016
* Wikipedia, JavaScript, [Online] Available at: https://en.wikipedia.org/wiki/JavaScript Accessed: 1 February 2016