JAVASCRIPT PART ONE GUIDE

This outline is a work-in-progress, and may change in the future – medatech@medasf.org

Mission Economic Development Agency | Instructor Eduardo Garcia

JAVASCRIPT

JavaScript is a computer language that allows us to provide *instructions* to a computer that can react when it is in a certain situation. Just like a cake recipe that you follow to make a cake, a computer follows the instructions in order to achieve a specific goal. But just like recipes, improperly written instructions are very unlikely to achieve the goal in mind! Please take a note that JavaScript and Java are not the same computer language, we have both JavaScript and Java, and while the code looks similar in both languages, they are not compatible with each other.

JAVASCRIPT SYNTAX & STATEMENTS

The JavaScript syntax is a little more complicated than HTML or CSS, but you generally want to take full advantage of indentation, spaces, and new lines. Every *instruction* for the computer is called a statement. A JavaScript statement ends in a semicolon (**;**) with a few exceptions. Take the following JavaScript statement:

```
var myFirstNumber = 100;
```

➔ The **var** word is a special keyword of the JavaScript langauge that specifies something. JavaScript has many different keywords that represent or do different things. In this context var means variable, and a variable in a programming language is a container that can hold a value. In JavaScript, a variable can hold any *single* value.

➔ **myFirstNumber** is the name of the variable, this is what we call the variable when we want to use it later. Notice that it does not start with a capital letter and there are no spaces in the name.

➔ **100** is the value that we store in the variable with the name myFirstNumber. If we use this variable later, we will get the value 100 out of it.

ORDER OF OPERATIONS

JavaScript works from the top to the bottom, just like reading an essay. Once JavaScript has gone through a statement it *generally* never returns to that previous statement to run it again. There are ways to specifically run a previous statement again but that will be covered later. For now, understand that the order of the instructions is based on when it first appears in your code. When statements use the single equal sign (**=**) or a pair a parentheses, then there is a more specific order on that statement. Everything on the right side of the equal sign must happen or run first. If there are a set of parentheses on the right side of the equal sign, everything in the parentheses must be *resolved* before continuing with the statement.

Example:

```
var myTotal = 10 – 100 + ( 3 – 2 );
```

1) The computer will first resolve what's inside of the parentheses.
2) The computer will then complete everything on the right side of the equal sign.
3) Notice that a variable can only hold a single value. So in this case myTotal is not holding a mathematical equation, instead it's holding the number **-89** when this statement runs.

STORING AND CALLING VALUES

Looking at the first code example in this document, we are creating a variable by using the **var** keyword and then assigning the name with a value. In computer programming, we store many different types of values within variables and ultimately have a result which is also stored in a variable. Take the following as an example:

```
var myFirstNumber = 100;

var mySecondNumber = 10;

var myFinalNumber = myFirstNumber + mySecondNumber;
```

Notice that in the third line we are not creating two more variables (after the equal sign), instead we are *calling* these two variables using their names. When we call a variable we are asking for the current value it is holding. In this case myFinalNumber is not holding two variables (variables cannot hold other variables) but instead it first calculated whatever myFirstNumber and mySecondNumber was holding and then storing the results in myFinalNumber!

THE FIRST THREE DATATYPES

In JavaScript we have different **Data Types** that represent a specific family of values. While there are more (a total of six as of 2019), we will be looking at three of the most commonly used data types in JavaScript:

- **String** A string is a collection of characters, these can include lowercase and uppercase alphabet characters, numbers, and special characters, including the space character! All Strings **must** be wrapped in single quotes or double quotes (but do not mix 'n match these).
  - "This is a string"
  - 'Another String!'
  - "100020" *note that this is still a String and not a Number data type.*
- **Number** A number is a data type that involves numbers, and as usual can only contain number characters, with the exception of the letter *e* that is used to describe exponents. Numbers **must not** be wrapped in single or double quotes.
  - 1000300
  - 10.93
  - -100
- **Boolean** A boolean is a special data type that represents the foundation of computer data, a *true* or a *false*. As computers can only have two states, on or off, true or false, 0 or 1, north or south, booleans allow us to quickly activate or disactivate code based on the current situation. In classical computer programming a boolean can only be true or false. The value should be all lowercase and **should not** be wrapped in single or double quotations.
  - true
  - false

NOTES