Chapter 2 Programing Basics

* Comments –
  + well-commented code is the hallmark of a ninja programmer. It makes it easier for anybody reading your code to understand what’s going on, and that includes you!
  + // Short comment
  + /\*\*/ Multi line comment.
* JavaScript Grammar
  + Similar to C programming language.
  + There’s no need to actually use a semicolon to terminate a statement because JavaScript interpreters use a process called Automatic Semicolon Insertion (ASI).
    - What if the code goes to long and needs two lines? How do we go about doing that? Like in JAVA would we do a + and space then continue on a new line? Or do we just write a really long line of code?
    - Then another thought when do we say a line/statement of code too long?
  + Blocks do not need to be terminated by a semicolon.
  + Whitespace (such as spaces, tabs and new lines) is used to separate the different parts of each statement.
* Reserved Words.
  + abstract, await, boolean, break, byte, case, catch, char, class, const, continue, debugger, default, delete, do, double, else, enum, export, extends, false, final, finally, float, for, function, goto, if, implements, import, in instanceof, int, interface, let, long, native, new, null, package, private, protected, public, return, short, static, super, switch, synchronized, this, throw, throws, transient, true, try, typeof, var, volatile, void, while, with, yield
  + the following words cannot be used as variables for they are already used by default things for JavaScript
  + undefined, NaN, Infinity -> these words are not reserved but should’ve been
* Primative data Types
  + String, Symbol, Number, Boolean, Undefined, Null
  + Any value that isn’t one of the primitive data types listed above is an **object**
    - Arrays functions, and object literals.
  + Any value that isn’t one of the primitive data types listed above is an**object**
* **Variables** 
  + **Refers to a value stored in memory**
* **Declaring and assigning**
  + From ES6 onwards, JavaScript uses the keywords const and let to declare variables. The keyword const is used when the variable will not be reassigned to another value, whereas let is used if the variable might be reassigned later in the program.
  + Variables that have been declared using the let keyword can be reassigned to another value at some point later in the program.
  + If the variable references a non-primitive data type, such as an array, function or object, then using const will not make it**immutable**. This means the underlying data inside the objectcanchange (known as**mutating**the object
  + Remember that a core tenet of the JavaScript language is that it has to remain backwardly compatible. This means that the behavior of var had to remain consistent, so couldn't just be changed in ES6. For that reason, the new keyword let was introduced.
  + The main difference was that variables declared using let and const have block scope, which is discussed in more detail below. They also prevent you from overwriting any built-in methods by assignment, which is generally frowned upon, whereas using var doesn't.
* Scope
  + Scope is an important concept in programming. It refers to where a constant or variable is accessible by the program. There are two common scopes that are often referred to in programs: global scope and local scope.
  + outside of a block is said to have **global scope**. This means it is accessible everywhere in the program.
  + **local scope**. This means that any variables defined inside a block using the let or const will only be available inside that block and not be accessible outside of that block.
* Converting Strings to Numbers.
  + The best way to change a string to a number is to use the Number method. This will convert the string form of a number into an actual number:
    - Number(’23)
* Converting Numbers to Strings
  + String(3)
  + 3+’’
  + 10..toString()
* Parsing Numbers
  + There is also a useful function called parseInt() that can be used to convert a string representation of a numerical value back into a number
  + This can be used to just select numbers from a string
  + To round up or move decimal points.

Chapter 3 Arrays, Logic and Loops

* Arrays
  + An array is an ordered list of values. To create an array literal, simply write a pair of square brackets:
  + To access a specific value in an array, we write its position in the array in square brackets (this is known as its index).
  + Each item in an array can be treated like a variable. You can change the value using the assignment operator =
    - After reading two chapters. I see writing in the console to look at certain things. How often does one do that when writing in javascript?
* Creating Array Literals.
  + We can create an array literal using square brackets that already contain some initial values, so there’s no need to add each value one by one
  + You don’t even have to use the same types of items inside an array.
  + The delete operator will remove an item from an array: delete avengers[3]
* Destructing Arrays
  + **Destructuring**an array is the concept of taking values out of an array and presenting them as individual values.
  + Destructuring allows us to assign multiple values at the same time, using arrays:
  + Destructuring also gives us a neat way of swapping the value of two variables over
* Array Properties and Methods
  + The length property is mutable, meaning you can manually change it:
  + To remove the last item from an array, we can use the pop() method
  + The shift() method works in a similar way to the pop() method, but this removes thefirstitem in the array
  + The push() method appends a new value to the end of the array
  + The unshift() method is similar to the push() method, but this appends a new item to thebeginningof the array
  + The concat() method can be used to merge an array with one or more arrays
  + . The spread operator is three dots, ... that are placed in front of an array, with the effect of spreading out the elements of that array. This can be used to spread the elements of two arrays and put them together in a new array, like so:
    - avengers = [ ...avengers, ...['Hulk','Hawkeye', 'Black Widow'] ];
    - << ['Captain America', 'Iron Man', 'Thor', 'Hulk', 'Hawkeye', 'Black Widow']
  + The join() method can be used to turn the array into a string that comprises all the items in the array, separated by commas:
  + The slice() method creates a subarray; effectively chopping out a slice of an original array, starting at one position and finishing at another.
    - The splice() method removes items from an array then inserts new items in their place
    - The splice() method can also be used to insert values into an array at a specific index without removing any items, by indicating that zero items are to be removed
  + We can reverse the order of an array using the reverse() method
  + We can sort the order of an array using the sort() method

Chapter 4: Functions

* A function can also be declared using the constructor Function()
  + It's not recommended to declare functions in this way as there are numerous problems associated with placing the function body inside a string
  + It's not recommended to declare functions in this way as there are numerous problems associated with placing the function body inside a string
* should always declare functions using function literals, function declarations or function expressions
* [Don’t Repeat Yourself,](http://en.wikipedia.org/wiki/Don%27t_repeat_yourself)or DRY, is a principle of programming that specifies that every part of a program should only be written once
* Every function has a special variable called arguments . This is an array-like object that contains every argument passed to the function when it is invoked. We can create a simple function called arguments() that will return the arguments object so we can see what it contains
* The problem is that arguments is not an array.
* rest operator. This was introduced in ES6 and can be used to deal with multiple arguments by creating an array of arguments that are available inside the body of the function = rest()
* ES6 introduced a convenient way to specify default parameters for a function. These are values that will be used by the function if no arguments are provided when it is invoked. To specify a default parameter, simply assign the default value to it in the function definition
* Hoisting is the JavaScript interpreter’s action of moving all variable and function declarations to the top of the current scope, regardless of where they are defined.
* Variable declarations that use the var keyword are automatically moved to the top of the current scope. Variable assignment is not hoisted, however. This means that a variable assigned at the end of a function will have a value of undefined until the assignment is made
* functions can also be given as a parameter to another function. A function that is passed as an argument to another is known as acallback.
* There is nothing to actually define a parameter as a callback, so if a function isn't provided as an argument, then this code won't work. It is possible to check if an argument is a function using the following code: