Ch 2 programming basics

* Comments in javascript // this a is a comment /\* this is a longer comment\*/
* Grammar is similar to c programming languages
* No need to use ae semicolon to terminate statement because JS interpreters use process called Automatic semicolon insertion (ASI) but you can still add them if you want
* Blocks are curly braces
* These are 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
* Objects include arrays, functions, object literals
* Typeof – ex: typeof {ninja: ‘turtle’} (return) << ‘object’
* Const is constant, let is if the variable might be reassigned later
* Exponential notation = 1e6 – 1000000, 2E3 – 2000, 2.5e-3 – 0.0025
* NaN – not a number
* parseInt() – can be used to convert string into number value

A screenshot of a cell phone

Description automatically generated

Ch 3 arrays, logic, and loops

* join() – can be used to turn array into string
* slice() – creates subarray, chopping out slice of original
* splice() – removes items from array, inserts new items in their place
* reverse() – reverses order of array
* set – data structure that represents collection of unique values
* ex: const list = new Set();
* ex adding: list.add(1);
* ex add multiples:m list.add(2).add(3).add(4);
* removing values: list.delete(2);
* maps – data structure, convenient way of keeping list of key and value pairs
  + similar to objects, maps can use any data type as key
  + creating map: const romanNumerals = new Map();
  + romanNumerals.set(1, ‘I’);

A screenshot of a cell phone

Description automatically generated

Ch 4 functions

* function can be declared using constructor, body of function then is entered as a string
* function hoisting – action of moving all variable and function declarations to the top of current scope = hoist();
  + //code here
  + Function hoist() {
  + Console.log(‘Hoist Me!’);
  + }
* forEach() – loops through array and invoke callback function
* colors.forEach( (color,index) =>
* console.log(`Color at position ${index} is ${color}`) );
* Reduce() – iterates over each value in array
* Filter() – returns new array that only contains items from original array