**UNIT 2 NOTES**

\*Document Object Model (DOM)

* shows the relationship between all HTML document elements
* defines elements as objects
* can be used with JS to add/delete/edit elements & attributes

\*Why put JS at the bottom? ->

to buy time for data retrieval (API) so the page doesn’t time out

but sometimes if you need a lot of data, you should put it in the head of the document

\*Characteristics in JavaScript are called properties

\*Variables hold a specific type of data (integer, float, string, character, boolean, null, object, etc.)

- variables are not ‘strongly typed’ (you don’t have to declare what data type it is)

- var variableName = value ; // equal sign is called the “assignment operator”

\*Arrays hold a list of items // each variable in the array is called an element -> each has an index #

Class exercise:

*var numberArray = new Array (7) ;*

*for (x=0; x<7; x++) {*

*var list = [1, 25, 36, 1000, -100, 0] ;*

*numberArray[x] = list[x] ;*

*}*

*numberArray[6] = 20 ;*

\*Operators = + \* - / % =

\*Comparison Operators === / !== / < / > / <= / >=

=== does an exact comparison / == tells if they’re the same type

\*Conditionals (If, if else, else if, etc.)

\*Output in JavaScript -

- innerHTML : allows you to set the HTML in a specific element

- recommended not to use with only text because it can cause security issues

- textContent : this can only set text and not HTML

- document.querySelector() : grabs an element based on the type that it is (i.e. “p”)

\*Functions (aka Sub-Routines) – basic way to make a function:

function myFunction (parameters) {

code to be executed using parameters ;

}

\*Loops – for, for each, while, do while, etc.