CSS

Selectors:

1. Class
2. ID (Ex: id=”hello”; In CSS: #hello{}
3. Attribute Selector:

Pseudo Elements: lets us style particular part of the selected element.

The Cascade: the order of declaring the styles matters.

* Hover attribute of button changes its property when cursor is placed over it
* !important>Inline styles>ID selectors>Class, Attribute and Pseudo Class selectors>Element and Pseudo Element selectors
* Inherit attribute: inherits the property of the nearest parent.

Position: static, absolute, relative, sticky

Transition, transform, scaling

JavaScript

* REPL[Read Evaluate Print Load]
* let var\_name=”value”
* const const\_name=”value”[values of constants don’t change]
* Booleans: true or false
* String length: str\_name.length
* String slicing: str.slice(low,high)

Template Literals: using the back tick(`) to create string then we can use ‘$’ as evaluater inside the string. Eg: `Hello 5+4=$(5+4)`

The Math object(or say the Math class/package) with various methods

Input: “prompt” command

Logical precedence: && > | |

Array operations:

* push and pop data from the end of the array. We can also push multiple elements at a time.
* Shift and unshift i.e delete and add from the beginning.
* Arr1.concat(arr2) joins two arrays into one.
* Indexof, includes, reverse
* Slice and splice
* Arr.sort()

Object literals in JavaScript

* Key – value pairs
* Object.keys and object.values to access the array containing all the keys and values respectively.

For…of loop: it’s just like using “for j in list” in python.

Ex: for(sub in subreddit){

Console.log(sub);

}

Random number generator: Math.floor(Math.random()\*range)+1;

Functions in JavaScript:

* Defining: function func\_name(x){ return x}
* Higher order functions: passing a function as argument in another function.
* Creating methods in JavaScript by putting more than one function in a single variable. Example: var\_name: methods, then methods.square(num), methods.cube(num); don’t use let before the functions created inside.
* ‘this’ keyword in JS is same as Java used to denote the current object.
* Try and Catch functions.
* Map method: creates a new array with the results of calling a callback function. Format: let var2=var1.map(function(y){ ……; return x;})