Day 4-7

Javascript (Not complete yet)

Running Javascript

1. Javascript can be run on different platforms that include
    a. Scrimba
    b. CodePen
    c. FreeCodeCamp's online editor
    d. HTML file just opened in a browser.

Commenting in Javascript

1. Inline comments are performed by adding // to the start of the line
2. Multi-Line Comment /* ...adsfadf… */

Data Types and Variables

1. Javascript has 7 defined data types
    a. Undefined - a variable that hasn't been defined yet.
    b. Null - nothing, set it to nothing
    c. Boolean - true or false
    d. String - a series of characters
    e. Symbol - a primitive value that is unique
    f. Number - number
    g. Object - stores key value pairs
2. Variables can be declared in 3 fashions
    a. Var - can be used throughout program

```
var myName = "Viswa"
```

    b. Let - can be used only in the scope of the place of declaration
    c. Const - can never change
    d. Blank - assumes global scope if declared inside the function (dependant on browser)

Storing Values with Assignment Operators

1. Use semicolon
2. console.log(variable) shows the value of a variable in the console.

```
var a;
console.log(a)
```

    Results in null
3. JavaScript variables are case sensitive.
4. Use camelCase, as it is generally preferred.

Arithmetic Operations

1. Same as usual

2. Incrementing and Decrementing is like in c++

Escaping String Literals in JavaScript

1. Using a backslash in a string makes sure that the actual literal is ignored.

```
var Str = "Viswa writes \"double quotes\"";
```

2. WHat one can do to escape both single and double quotes is to use back-ticks at the starting or the ending of the string.

```
var Str = `Viswa writes "double quotes"`;
```

3. The various things one can escape in js
    a. \' - single quote
    b. \" - double quote
    c. \\ - backslash
    d. \n - newline
    e. \r - carriage return (Cursor moves to beginning of line)
    f. \t - tab
    g. \b - backspace
    h. \f - form feed (Cursor moves to the next page)

Strings

- Concatenation is similar to python
- Length function is .length
    - var variable = "Viswalahiri"
      lastNameLength = lastName.length;
- Strings are immutable, meaning they can't be altere, even though they can be changed.
- One can gain access to the last letters in a string by using the bracket notation and .length -1 as the index

Arrays

- Arrays allow to store several pieces of data in a single place
- Begin and end with a bracket
- Values separated by commas
- Same syntax as Python
- Arrays are mutable
- **Nested arrays**
    - Also known as multi-dimensional arrays
- **Manipulating Arrays**
    - push()
        - Append data to array
        - Append either value or array (very much similar to Python)
        - Arr = [`Viswa`,`Lahiri`]
          Arr.push(`H`)
          => [`Viswa`,`Lahiri`, `H`]
    - pop()
        - Remove last element from array, and assigns it to a variable.

    - shift()
        - Removes the first element instead of the final element.
        - Similar to pop()
    - unshift()
        - Adds element to the beginning of the array, similar to push().
        - Similar to push()
- **Functions**
    - Allow reusing code

```
function viswalahiri(){
  console.log(`Hrrmmm. Strong the force is. Yes, hrrrm.`);
}
viswalahiri();
```

    - **Parameters**
        - Parameters are passed to the function like in any other language
    - **Scope**
        - Variables outside the function, have a global scope.
        - Whereas those inside, have local scope

```
function viswalahiri(){
  var1 = 10
  var var2 = 10
}
```

          In the above function var1 is global, and var2 is local (dependant on platform)
        - Whereas those inside, have local scope
        - Local > global (precedence)
- **Queue**
    - **Note:** JSON.stringify(arr) easily prints an array in a string format to the console.

```
function queue(arr,item){
arr.push(item);
return arr.shift();
}
```

- **Booleans in Functions**
    - return true
    - return false are the two returns

- **Conditional Statements**
    - Similar to python apart from elif being else if

```
switch(val){
case 1:
    Do_something;
    break;
case 2:
case 3:
    Do_something;
    break;
default:
    Do_something;
    break;
}
```

- Equivalence Operators
    - '===' is the string equivalence operator, and it checks if both are equivalent without any type conversion
    - '==' is the equivalence operator and performs type conversion in order to bring them to a common type. It then checks if both are the same.
    - '=' is not an equivalence operator but it is an assignment operator
- Inequivalence Operation
    - '!==' performs no type conversion
    - '!=' performs type conversion
- Logical Operations
    - '&&' - and
    - '||' - or