**1.export and import: allow user to import the exported function from other file (make project more managabble)**

The javascript with in th html

<script src="path"></script>

**1.Type( weak type) in javascript**

#path

let variabe = value;

const variabe2 = value2; //Can't be overriden

var variable3 = value3; //The old way of defining variables

/\* multipline comments \*/

// reference types

arrays

functions

dates

**2.Variable types in javascript:**

variable = Type( variable)

typeof variable

nonempty string is true!

**Number**

a\*\*(expoenents)

B I D M A S -> the sequence of math operators

**String**

let a = "abc";

let b = "def";

let ab = a + b; // concatenation

let a1 = a[0]; // getting charactors

let aLength = a.length //getting the length of the String

.toUppercase

.indexOf('String')

.LastIndexOf('String')

.slice(fristIndex, lastIndex)

.substr(fristIndex, NumberOfCharactorsAfterFirstIndex)

.replace("targetString","replacedString")

`This is the formated string ${variable}. This is the template String`

**Boolean**

let a = "abc";

let test = a.include("a") // return true

console.log(a === "abc") // strict comparasion

**Null (a varaible with no value)**

**Undefined (for variables that have not yet been defined)**

**Object complext data structures**

/\* arrays related\*/

let ninjas = ['element0','element1'];

ninjas.concat(['element2'"])

ninjas.push('element3') //return the length of the pushed array

ninjas.pop() //return the last element in the array

**Symbol used with objects**

**For Loop/ if Loop/ switch**

/\*same as the java grammar.\*/

//for

for (let i = 0; i < 5; i++) {

console.log("Number:", i);

}

//if else

let age = 18;

if (age < 18) {

console.log("You are underage.");

} else console.log("You are an adult.");

//switch

switch (expression) {

case value1:

// Code if expression === value1

break;

case value2:

// Code if expression === value2

break;

default:

// Code if no cases match

}

**4.Function**

function functionName(parameters = defaultValue) {

//This way of defining leads to hoisting

//which compiler would first store all the unassigned function first!

return result;

}

const functionName = function(parameters = defaultValue) {

// Code to be executed

};

const functionName = (parameter) => {

//Code tobe executed

}

//if we have only need to return a value we can write

const functionName = (parameters) => returnValue;

/parameters could also be functions

Callback using built-in array

array.forEach(callback(element, index, array))

**5.Object**

let user = {

name : '123',

age : 123,

blogs :[1,2,3,4],

login : function(){

console.log('you are th best');

}// when using this inside the () = > arrow function

// js treats it as the windows objects! So we can only use it

//in the real functions.!

}; //just like the dictionary in python

user.name = 'new one';

user['name'] = 'the newr one';

**6.Interacting with a Browser**

const variable = document.querySelector('tag.className' & 'parentTag > childTag');

const allVariables = document.querySelectAll('tag or other');

allVariables[x], allVariables.forEach(variabels => variables.attributes);

const elementVariable = document.getElementById('idName');

const elementVariable2 = document.getElementByClassName('className');

const elementVariable3 = document.getElementByTagName('tagName');