

Jscript used:

Node.js is a complete framework which is used fo rthe server side web development used to make dynamic web app.

Similar to node.js , we have React which is the front end lib developed by facebook initially where we can create web as well as mobile app. With help of this ,writing the same code ,which can be run on both web as well as mobile .

React native is completely used for mobile app like IOS n android.

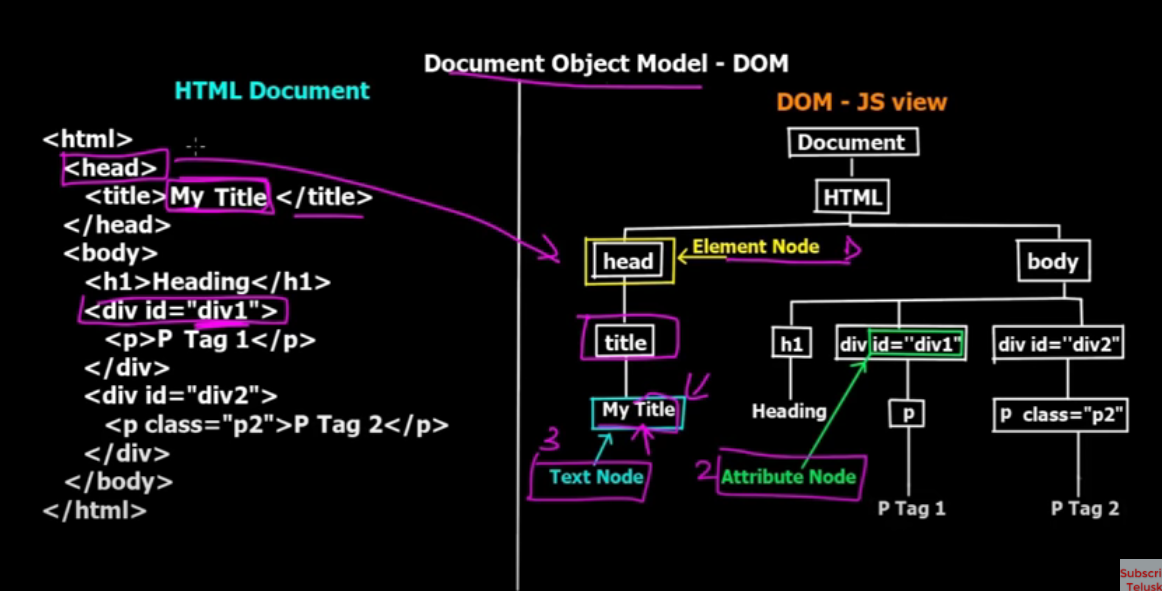
Angular JS which Is another MVC (Model view and controller) framework.Thi swas initially developed by google and now is open source used to develop dynamic web app.

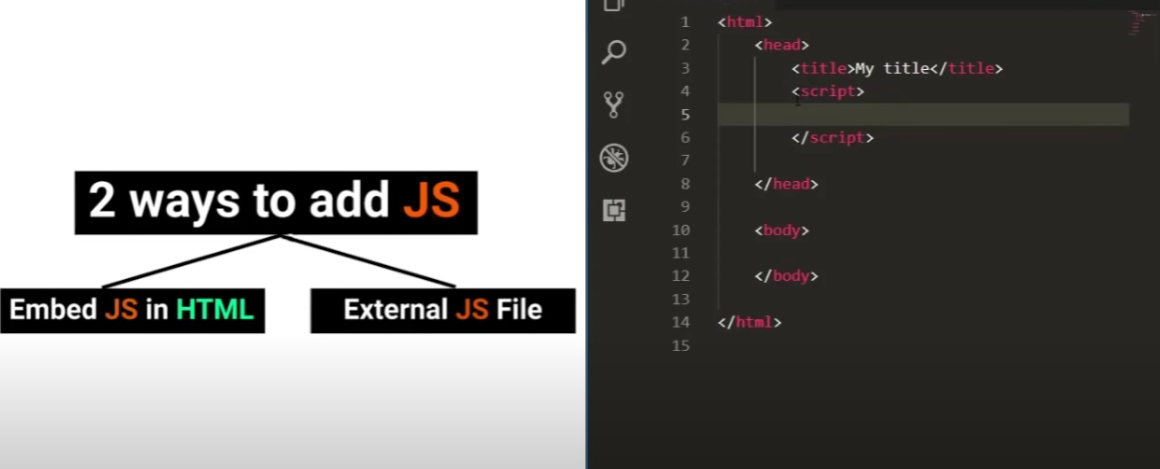
Desktop app: Electron : used to build cross platform apps for standalone systems (text editor , music payer etc).

React VR(virtual reality) : using this we can create VR experience,

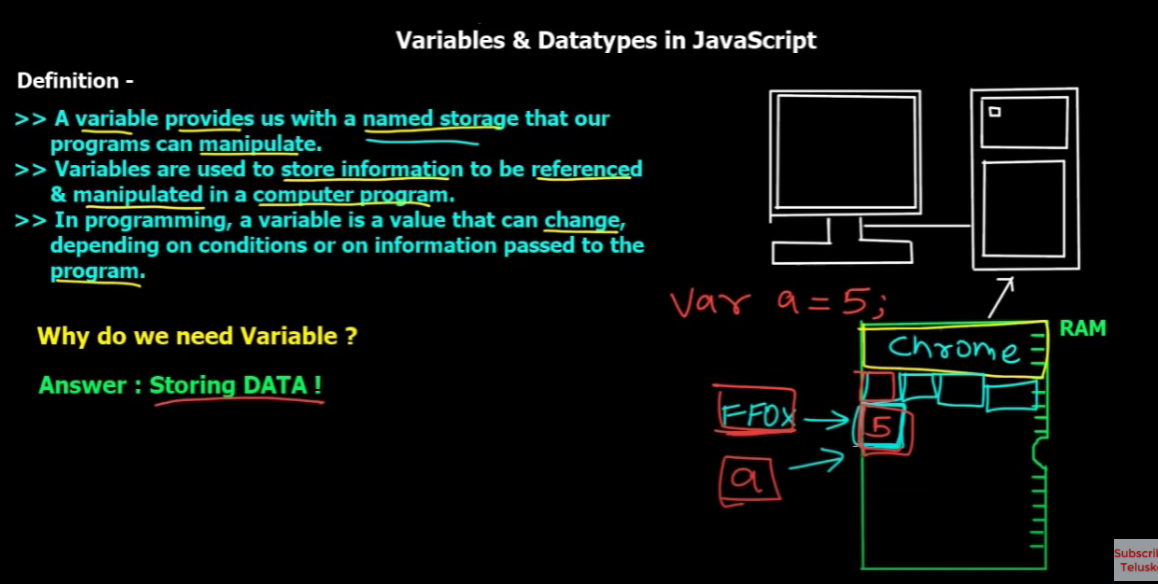
TensorfFlow.js is an open source machine leaning platform can be used for training and deploying machine learning models

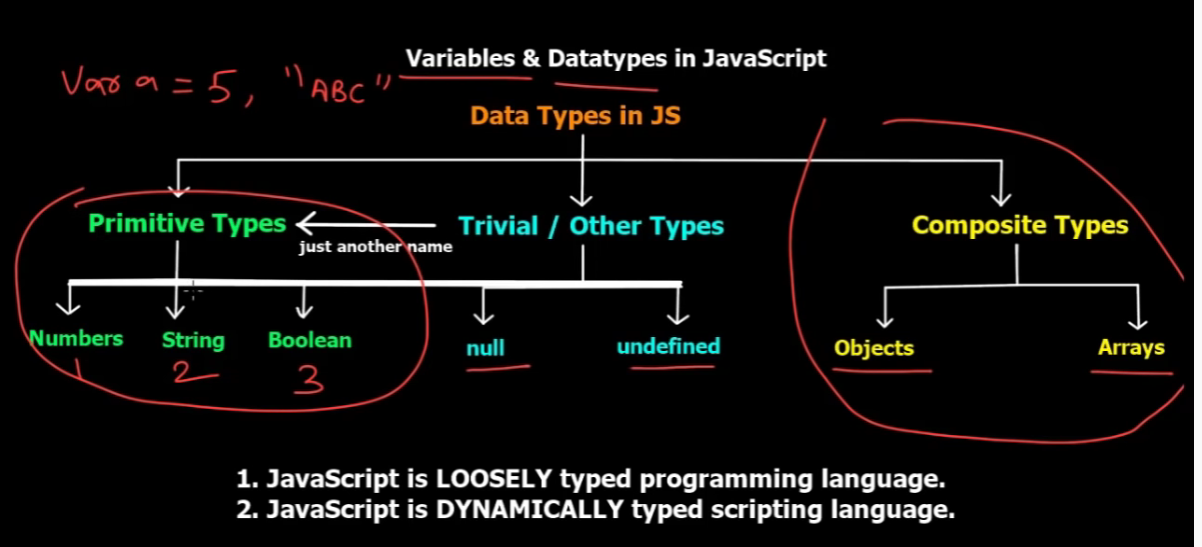
DOM : Document Object model :

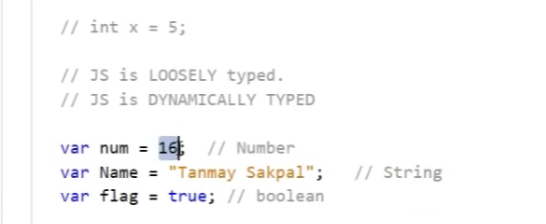




Got o extension in the VSCode , and install the live server.







Jave script is loosely types language coz we just declare any variabel be it number or string as

Var a= 5;

Or

Var a= “Name”;

Its also dynaically types as it changes the value dynamically without any erorro

//datatypes

let name1="Fouzia";

let age = 30;

let color = 'red';

console.log(name);

console.log(color);

const range = 3;

//range = 5;

console.log(range);

const numberOfDays=7;

let s1=" Hello WOrld!!";

let num= 18;

let salary=239.23;

let flag= true;

let fj=false;

let price;

let model= null;

let wheels = undefined;

Dynamic typing:

In dynamic typing , the variables initially declared as integer can be reassigned to the string value;

let num1 =98;

num1="Inu";

Reference types :

Objects

Arrays

Functions

objects:

Objects are defined as below: for user in flower brackets

Let user ={ } ; //This is undefined object with nothing defined.

If we want to define the properties in object then we need to define the name and value with semi colon

let user={

    name:"Inu",

    age:3

};

console.log(user);

console.log(user.name + " "+ user.age);

//using dot notation

let user={

    name:"Inu",

    age:3

};

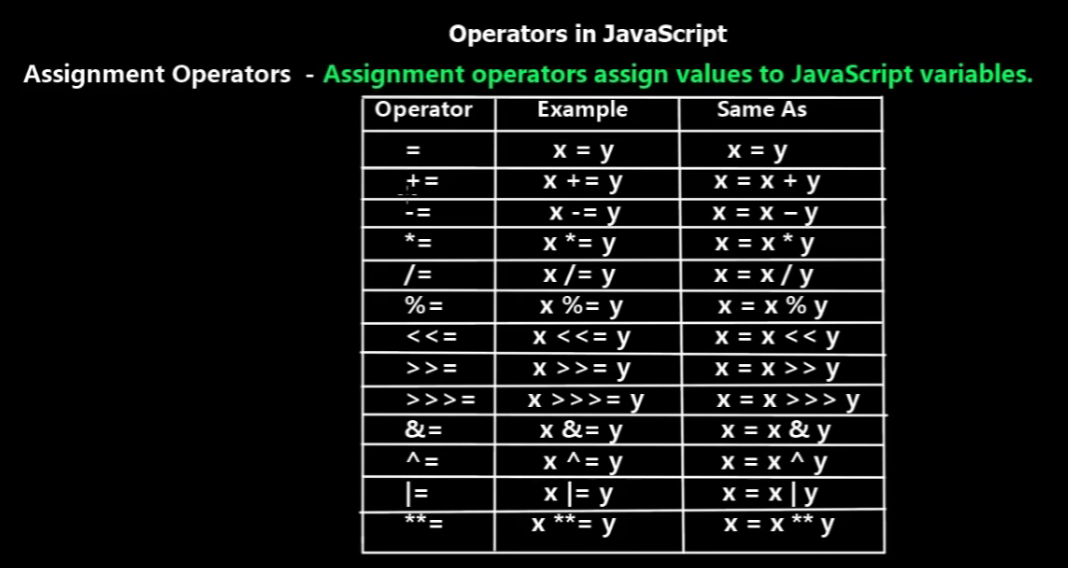
console.log(user);

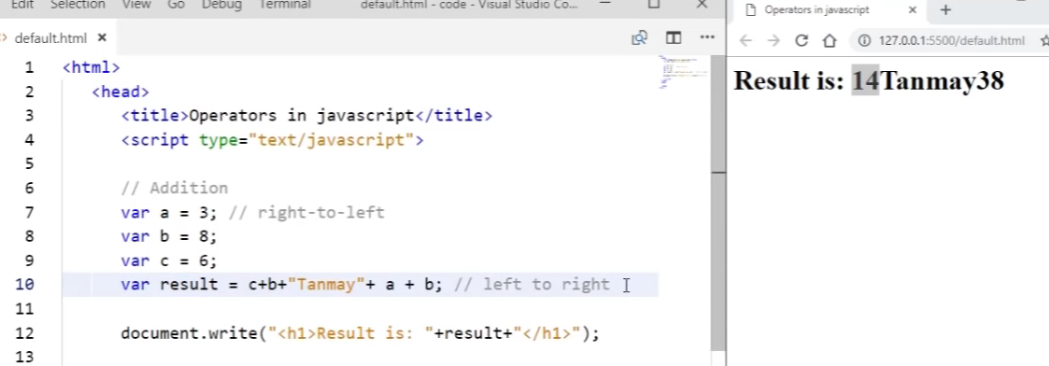
console.log(user.name + " "+ user.age);

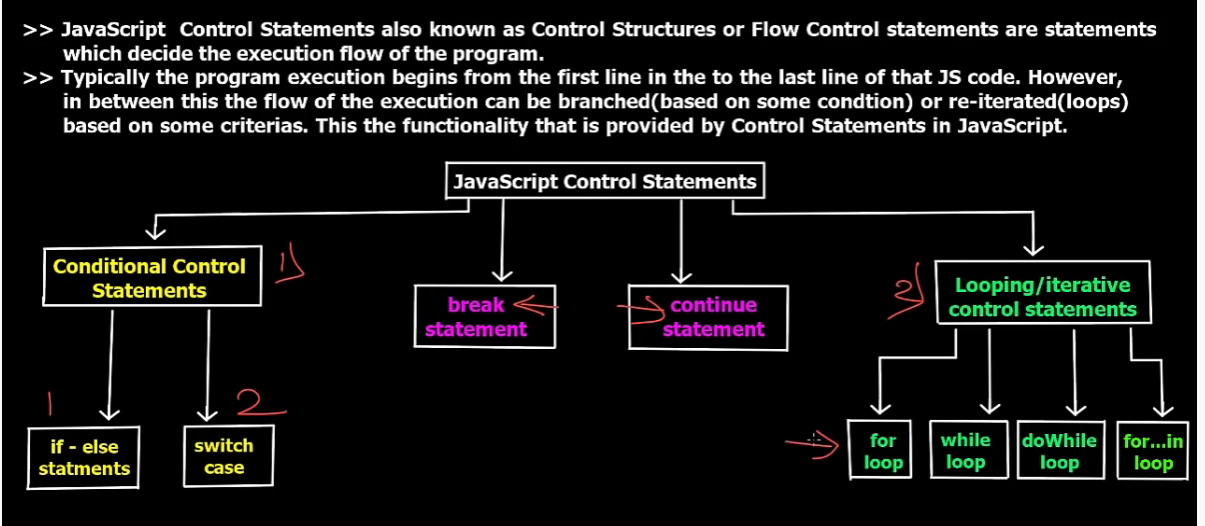
//using bracket notation

user['name']= "Alee";

console.log(user.name);







Array :

This is an empty array :

let language=[];

array with variables

let language=['java','python','Ruby/'];

console.log(language);

console.log(language.length);

language[3]='Selenium';

console.log(language[2]);

and if we give the value of array which is not there it wont give an array out of biund exception like java but it will simply give undefined.

Function:

With zero param

//zero parameter function

function getName(){

    console.log("This is my Function!!")

}

With parameter:

//param based function

function greet(name2,age)

{

    console.log("Hello " +name2+ " "+age);

}

greet("Fouzia",20);

greet("Inu")

If we don’t enter the second parameter it doesn’t give any error but says undefined

Operators:

Arithmetic

Assignment

Ternary

Comparision

Logical

Binary

//Arithmetic

let x=20;

let y= 10;

console.log(x+y);

console.log(x-y);

console.log(x\*y);

console.log(x/y);

console.log(2\*\*4);

//++ and --

let p=1;

let q=p++;

console.log(p);

console.log(q);

let p1=1;

let q1=++p1;

console.log(p1);

console.log(q1);

let m=7

console.log(--m);

console.log(m);

//assignmnet

x=5;

x+=20;

console.log(x);

//comparision

let x1=10

console.log(x>1);

console.log(x1===10);

console.log('11'===10);

console.log('10'==10);

console.log(true==1);

=== is called strict equality -which compares tyoe as well asvalue

== is called lose equality ehichc ompares only value be it s character or number.

Ternary operator:

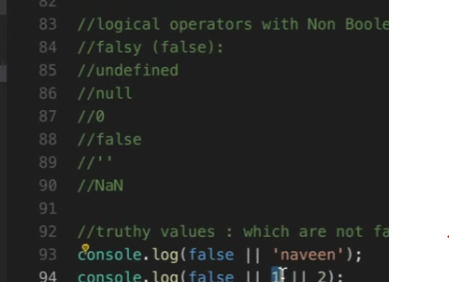
//ternary

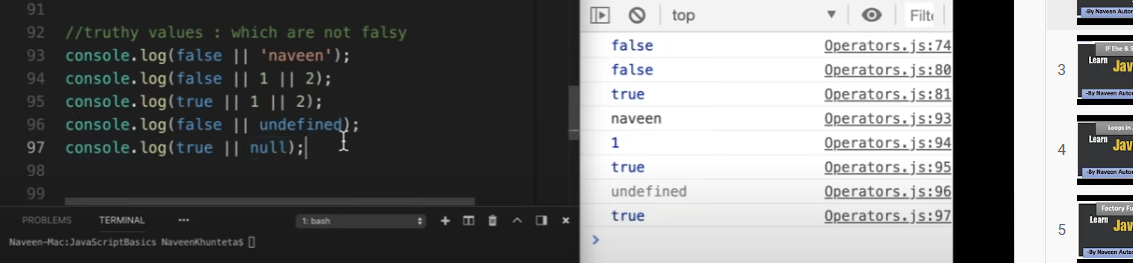
let points=10;

let type= points > 100 ? 'gold' :'silver';

console.log(type);

Ternary operator Is the question mark with the two values with :





Crearing single object :

//vars and functions should be defined in an object

//key : vlaue pair format

//in this corcle we have two variables, one object -location and one drawfunction

const circle={

//defining properties inside the braces

radius: 7,

isVisbile : true,

//we can describe onject inside the object with key n value pair

//here location is another object witin the circle

location : {

    x :1,

    y:1

},

draw : function(){

    console.log('draw');

}

}

console.log(circle.radius);

console.log(circle.location);

console.log(circle);

but this is not applicable if we are creating 100 of circles like this .

That’s why we need to create the factory functions : which produces the objects.

Name: camel case createCircle

//factory function: it produces the objects

function createCircle(radius){

//     const circle ={

//         radius : radius,

//         draw(){

//             console.log('draw function');

//         }

//     }

//     return circle;

// }

//instaead of the avbove code we can minimize the code as below

return{

    radius,

    draw(){

        console.log('draw function');

    }

}

}

const c1= createCircle(3);

console.log(c1);

c1.draw();

const c2= createCircle(8);

console.log(c2.radius);

constructor functions : also used to create objects

here we use the pascal notation :OneTwoThreeFour

//constructor Function : also used to create objects

//Name: pascal notation : OneTwoThreeFour

function CircleTwo(radius){

this.radius=radius;

this.draw=function(){

    console.log('draw function', radius);

}

}

//how to call constructor function : using new keyword

const a= new CircleTwo(6);

console.log(a);

console.log(a.radius);

a.draw();

The difference between the factory function and the constructor function is that we are using the new keyword and also this keyword for the variables as well as functions.

Dynamic objects:

//dynamic nature of objects in Jscript : at run time after creating

//the objects we can simply add the properties nd delete the prop

const circle ={

radius :1 ,

color :'red',

log(){ console.log("This is log function ",this.radius);

}

}

console.log(circle);

//add more properties in circle...dynamically

//there is no add keyword for adding the prop but for

//delting we write delete keyword

circle.price=100;

circle.isPresent=true;

circle.draw =function(){}

console.log(circle);

//delegte properties from circle object

delete circle.price;

console.log(circle);

delete circle.draw;

console.log(circle);

circle.log();

Value type vs reference types.

//value type : number, string,boolean,symbol(VS6),undefined,null

//ref type : Objects, functions, arrays

//primitve data types are copied by their values

let x =10;

y=x;

x=20;

console.log(x);

console.log(y);

//declaring objects

//here objects-non -primitive data types are copied by their references and not values

let pObj={value:10}

let q=pObj;

pObj.value=20

console.log(pObj.value);

console.log(q.value);

Enumerated objects:

//enumerating properties of Object

//get all the properties of the object

const circle={

    radius :3,

    draw(){

        console.log("This is a Draw function!");

    }

};

//1. for- in loop

for(let key in circle)

console.log(key,circle[key]);

//console.log(circle[key]);

console.log('========');

//for -of loop

//here for(let key of circle) will not work coz this works only

//for arryas and map thats why for the other we need to mention object.keys

for(let key of Object.keys(circle))

console.log(key);

for(let entry of Object.entries(circle))

console.log(entry);

//3. in opereator

//chk propert in object

if('radius' in circle)

console.log('Yes Radius is available!')

console.log('========');

if('price' in circle)

console.log('Yes Price is available!');

console.log('========');

if('draw' in circle)

console.log('Yes draw is available!');

console.log('========');

Clone objects :

//copying the one object o another object.

const circle= {

    radius : 1,

    draw(){

        console.log('Draw function');

    }

}

//1. using for - in loop

const object2 ={};

console.log(object2);

console.log('===after cloning====');

for (let key in circle)

{

    object2[key]=circle[key];

}

console.log(object2);

//2. using object.assign

console.log('===after cloning using obj assign====');

const object3=Object.assign({},circle);

console.log(object3);

//already assign values instaed of blank

console.log('=== cloning using obj assign existing prop====');

const obj4=Object.assign({ color :'red'},circle);

console.log(obj4);

obj4.draw();

//3. using spread ...   (three dots) operator

//here ... will behave like a blank  object and store all prop of

// circle into it

console.log('=== cloning using ... dots/====');

const obj5= {...circle};

console.log(obj5);

Template literls : (ES6)

const mssg='This is my first Java Script Code';

const mssg1='This is my\n first C Script Code';

const mssg2='This is my\n'+

                    '\'first\' C# Script Code';

console.log(mssg);

console.log(mssg1);

console.log(mssg2);

//literals

//object ={}

//Boolean =true,false

//String ='' ,""

//Es6 -template literals --> BAck tick chars --> ``(on keyboard it before 1)

//Here the aleena n fouzia name is hard coded , so we can write any name and then

// append it . by adding ${}

let product =function(){

    return 'Apple';

}

let name =  'Aleena';

let name1 = 'Fouzia';

const mailBody=`Hi ${name} ${3 + 2} ${product()};

This is ${name1} !

Thanks for adding me in the mail list.

HAve a good Day !!

Regards :

Fjaveed.`

console.log(mailBody);