MY JS notes

# Datatypes

# 

1 what is the difference betwn null and undefined?

2 what is typeof operator in js?

3. what is the result of typeof null in javascript?

4. what are primitive datatypes in javascript>

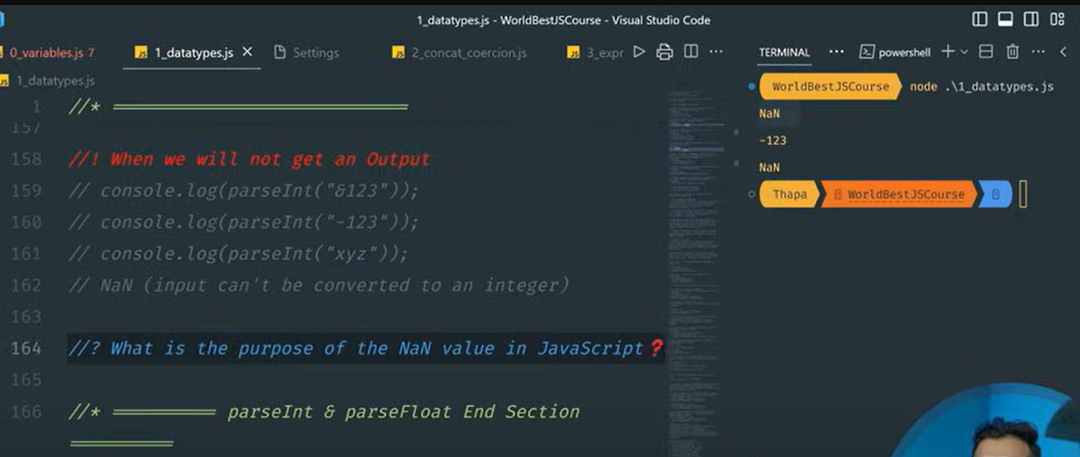
5. convert a string to number convert a number to string

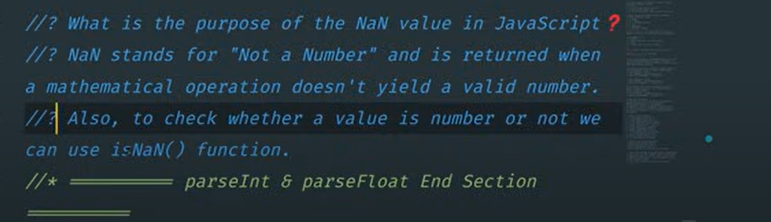
6. concept of truthy and flasy values in javascript

7 to checkmif a non-empty string in javascript is truthy or falsy in javascript we use if statement?

8 what is parseInt and parseFloat?

9 what is NaN





Console.log(isNaN(“Vinod”));

o/p: true’

console.log(isNaN(5));

op: false

9: why NaN== NAN is false?

10. what is the purpose of NaN value in javascript?

===================================================

# Concatenation in javascript

* Concatenation using + sign
* Type coercion: implicit and explicit coercion
* Implicit happens automatically

Ex: let sum = “5”- 5 or let sum = ”vinod” - 5

Console.log(sum) Console.log(sum)

o/p: 0 o/p: NaN

* Note: if we use + between a string and a number it is implicatily performed as concatenation.
* Let sum = “5” + 50 let sum = “5”- 50
* Out put is: 550 output is : -45
* Explicit needs to be done manually

Ex:

Imp:

console.log(10 + "20") // 1020

console.log(9 - "5") // 4

console.log("Java" + "Script") //javascript

console.log(" " + " ") //

console.log(" " + 0) // 0

console.log("vinod" - "sai") // NaN

console.log("false" + true) // ture (1+0)

console.log("4" - 2) //2

console.log(true + false); //true (1+0)

console.log(false + true); //true

console.log(false - true); // -1

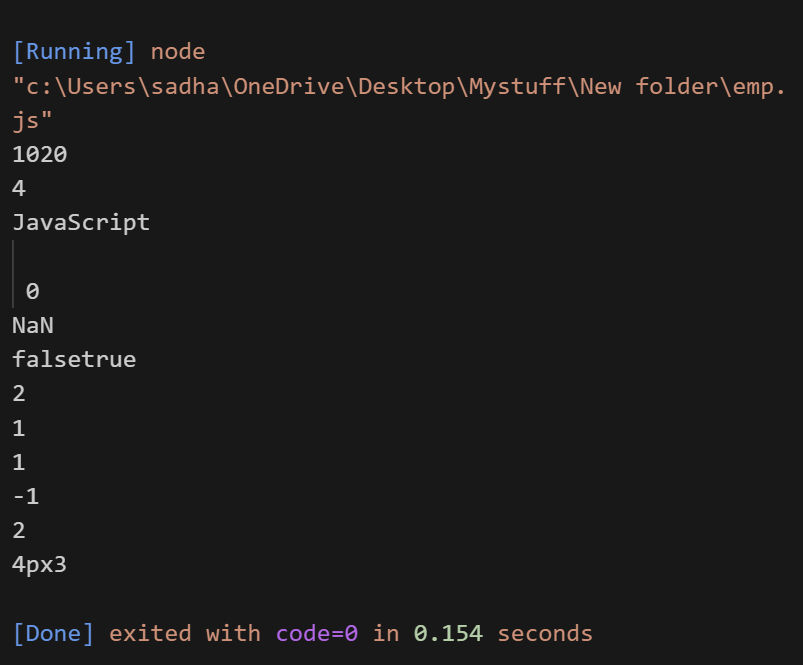
console.log(true + true); //2

console.log("4px" + 3); //4px3

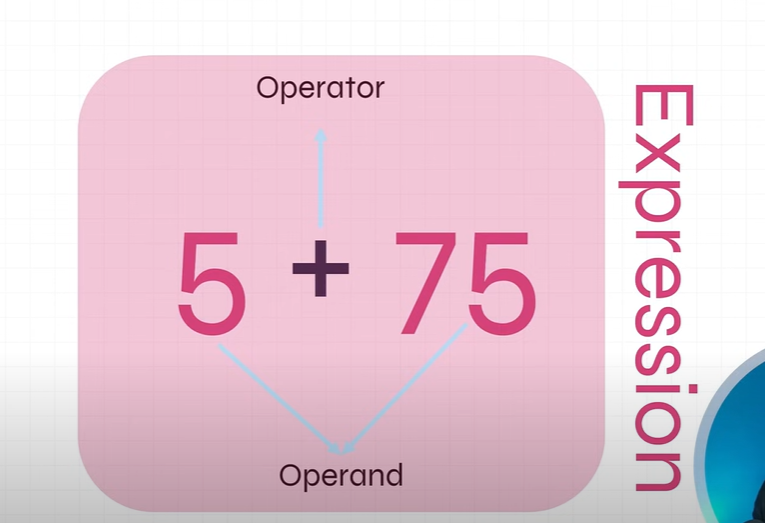
console.log(" -9" + 5); // -95

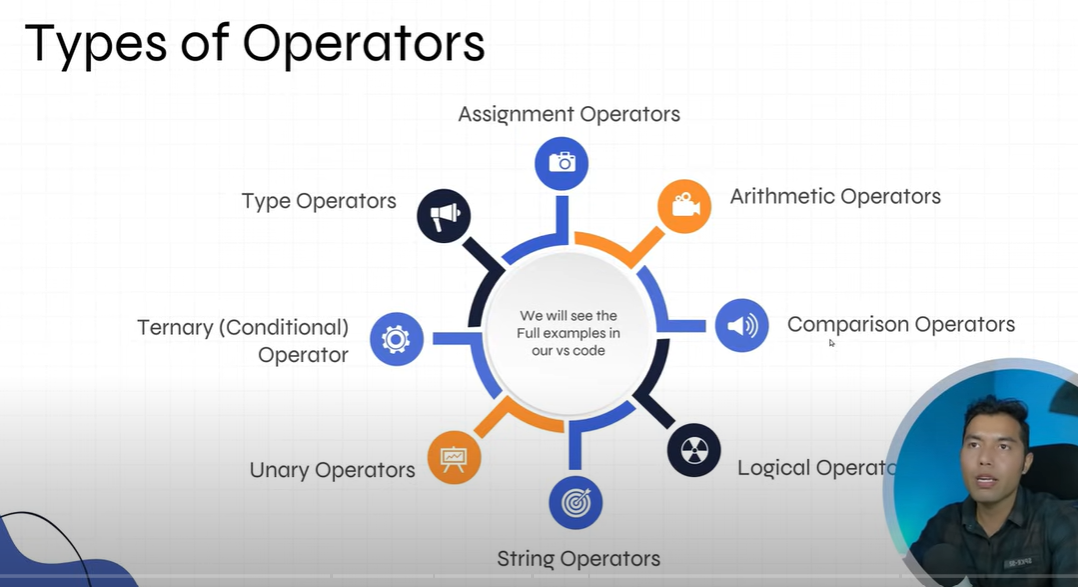
let result = “hello”/ 2 //nan

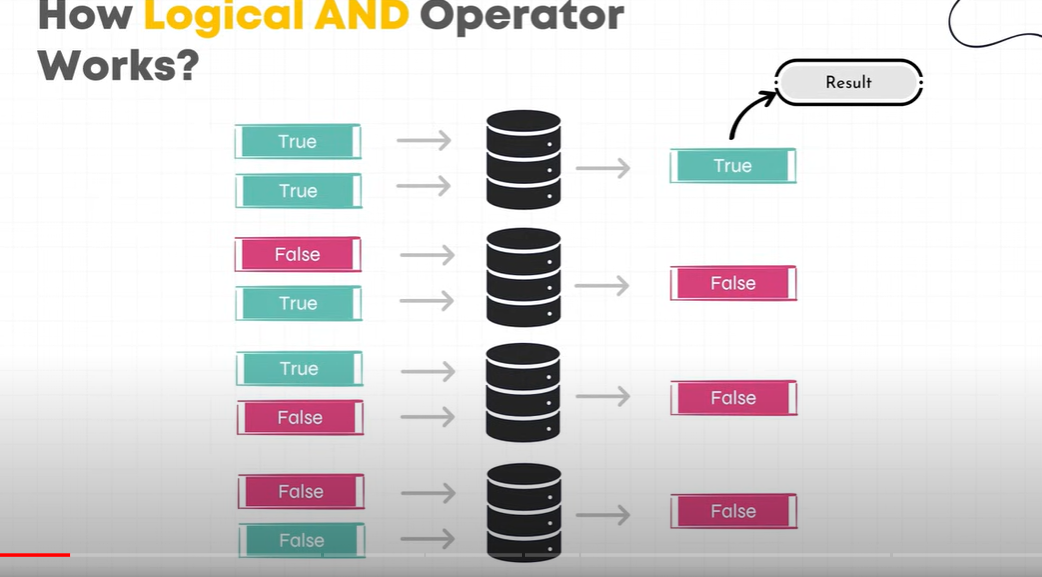
let sum = 0.1+0.2 // 0.300000000004

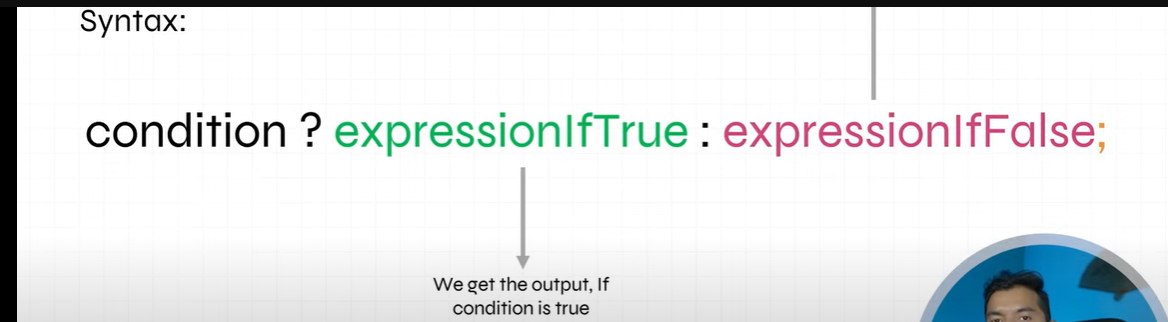


# Expressions and operations









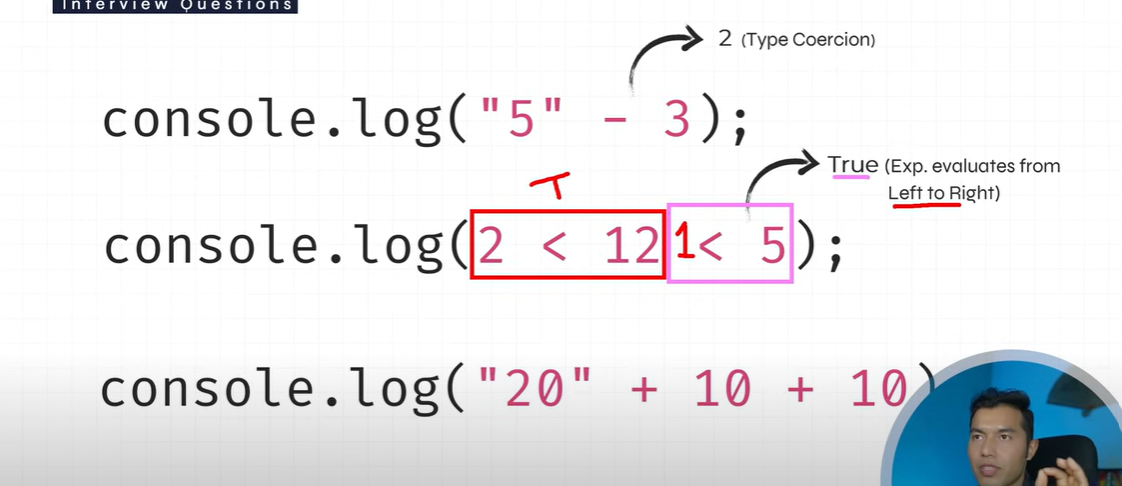
1 whats the result of of : var result = 0.1 + 0.2

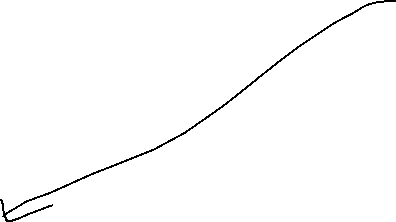
**Console.log(result.toFixed(2))**

**// toFixed gives the only two decimal values**

**Note: JS is case sensitive**

2. diff between == and === oerators ?

3. 



Note: The calculation starts from **left to right** so,



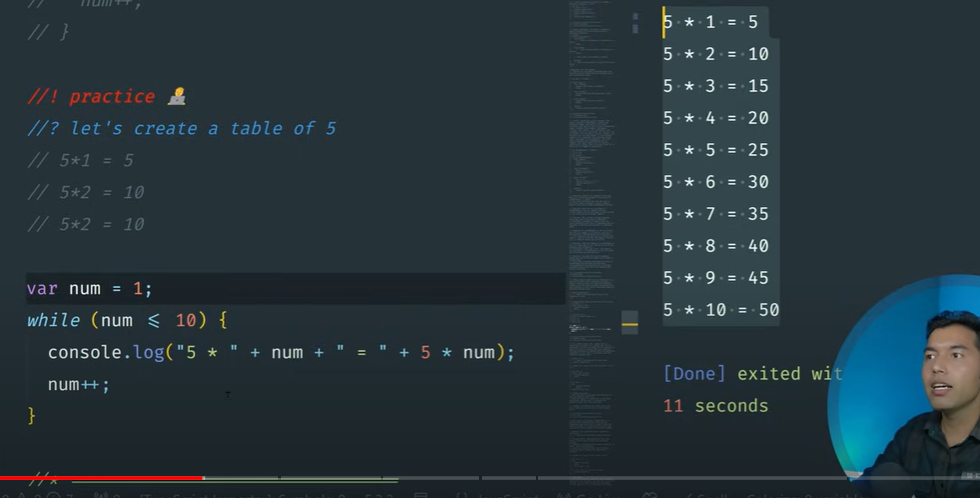
first **2 is compared with 12 :** hence **2< 12** : the result **is true**

Now the **true holds value 1** in JS : hence **1< 5 : true** , so the final result is **true.**

**Note :** left to right calculation

**“20” + 10 = “2010” +10 = 201010** because **(String+ number = string)**

# Control statements and loops



Ex: Var num =1;

While (num <= 10)

{

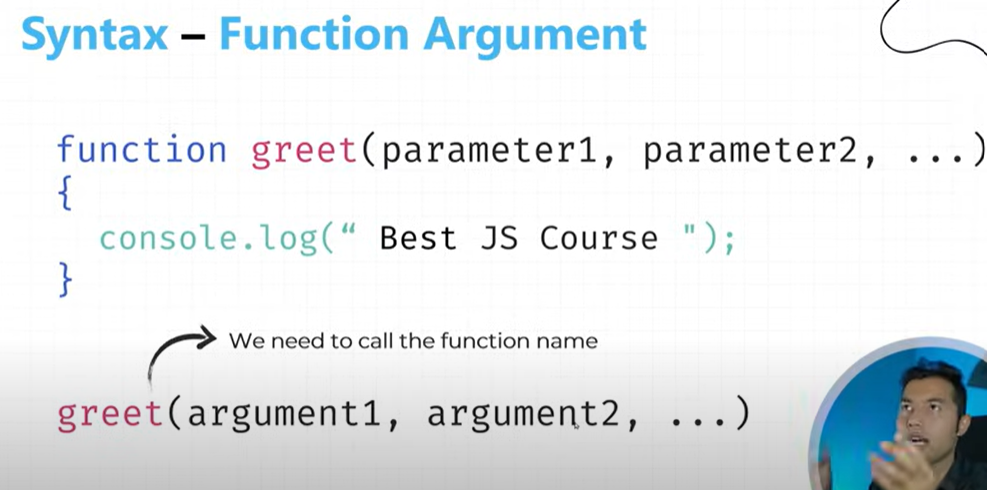
Console.log(“5 \*” + num + “ = “ + 5 \* num);

Num++;

}

======================================================

# JavaScript Functions



Note : String and variable must be printed using conactination as shown below.



Ex:  function greet(name, place){

    console.log('Hello  '+  name + ' from '+ place +' welcome to javascript');

 }

greet("ram", "hubli");

Anonymus function or function expression



Ex: let result = function (a,b){ // function with no name is assigned to



Console.log(a+b) //the variable so calleanonymus function

}



Result (5,10)

Retrun keyword

Anthing written after a return keyword in a function is ignored

Ex: function sum (a,b) or let result = function (a,b) {

{ retrun a+b;

Retrun (a+b); }

} result(5,10);

Sum(5,10);

IIFE : Immediately oinvoked function expression

(Function (a,b){

Console.log(a+b);

})(5.10);

(function (a,b){ // output is The sum pof a+b is 15

    console.log("The sum pof a+b is "+ (a+b));

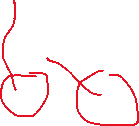
})(5,10);

===================================================================

ECMA Script 2015 : ES6

1. Let and const : let (block scope ) mutable, const(block scope ) immutable
2. Template strings
3. Default arguments
4. Arrow function





// the output will be vinod and undefined

Note : use ctrl+ backtick (`) to open the terminal

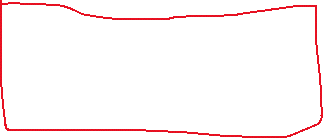
# **Template string** ``

**Ex : String interpolation**

**let age=33; // using template string such as**



**let message=`I am ${age} years old`; backticks and ${}, we can access the**



**console.log(message); value of the variable.**

===============================

Also we can print multiline output

let result =`

hello

my name is sadhana

located in hubli`;

console.log(result);

===============================

//output appears in multine

hello

my name is sadhana

located in hubli

=====================================================================

let num=1;

for(num==1;num<=10;num++){

console.log("5 \*"+num+"="+5\*num);

// console.log('5 \* ${num} = ${5\*num}');

}

// using template literals

**Default parameter**

**Function sum(a=10, b=10){ default prameter**



**return a+b;**

**}**

**Console.log(sum(5,10)); Actual paraemter**



Note : the programs considers the actual paremeters always, the default parameters are executed only when actual parameters are not passed.

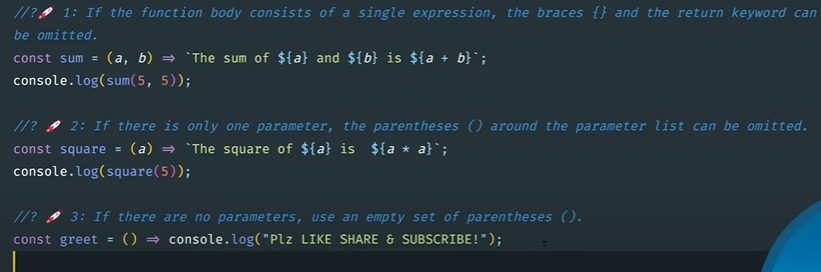
**Fat Arrow function**

# Normal function

Let sum = function(a,b){

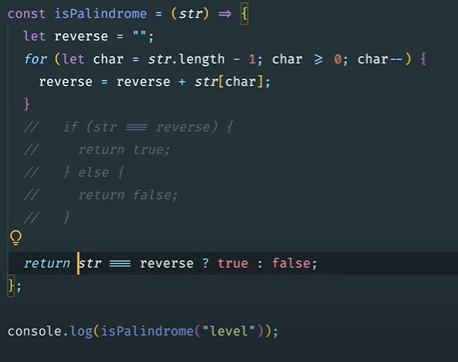
Console.log(“the sum of two number is a+b”);

}sum(1,2);



1. Program to write a calculator
2. Program to reverse a given string
3. Program to check
4. Program given string is palindrome or not

==============================================



Array

Array in JS is an **object : represents a collection of similar type of elements.**

**Each element can be accessed by index values**

Iterable : for of loop

# Creating array in three ways

Arry literal

Let arr=[‘apple’, ‘bana’, ‘orange’]

Console.log(arr);

Array constructor

Let arr=new array(‘a’,’2’,’3’)

Console.log(array);

Empty array

Let arr=[];

Console.log(typeof arr);

# Loops

1. For of
2. For In
3. ForEach

array.ForEach((currentvalue,index,array)=>

{

// write your logic

}, this value);

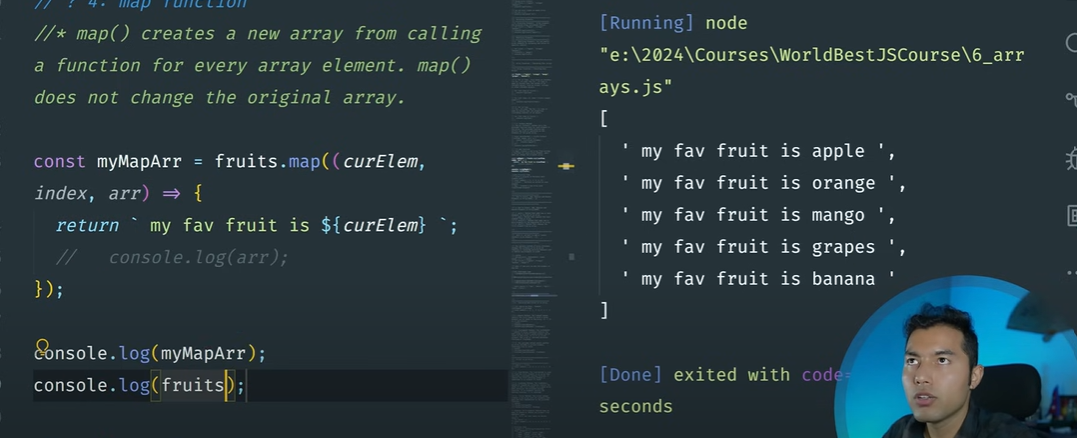
Note: return keyword cannot be used with foreach method

1. Map function

Array.map((currentvalue,index,array)=>{

//write your logic

}. This value);



let numbers=[1,2,3,4,5];

 numbers.forEach((curElement,)=>{

    console.log(`${curElement} ${curElement\*2}`);

 });

 console.log(numbers);

//  console.log(`${myNum}`);

//  console.log(myNum);

Running] node "c:\Users\sadha\OneDrive\Desktop\Mystuff\New folder\arr.js"

1 2

2 4

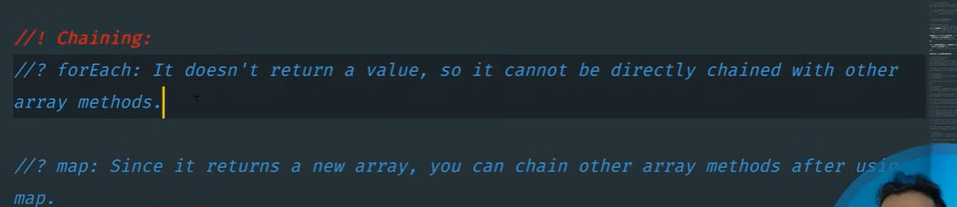
3 6

4 8

5 10

[ 1, 2, 3, 4, 5 ]





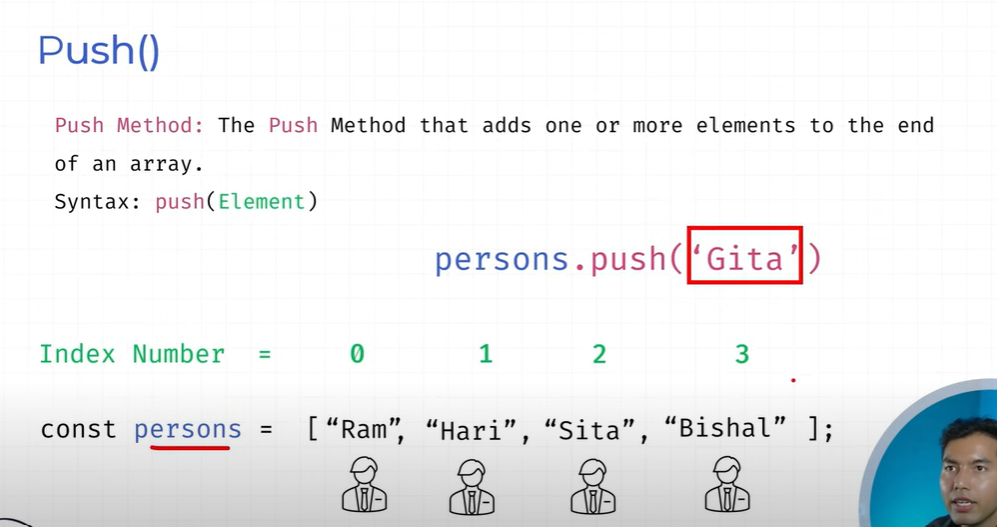
# 

================================================================

# Array methods

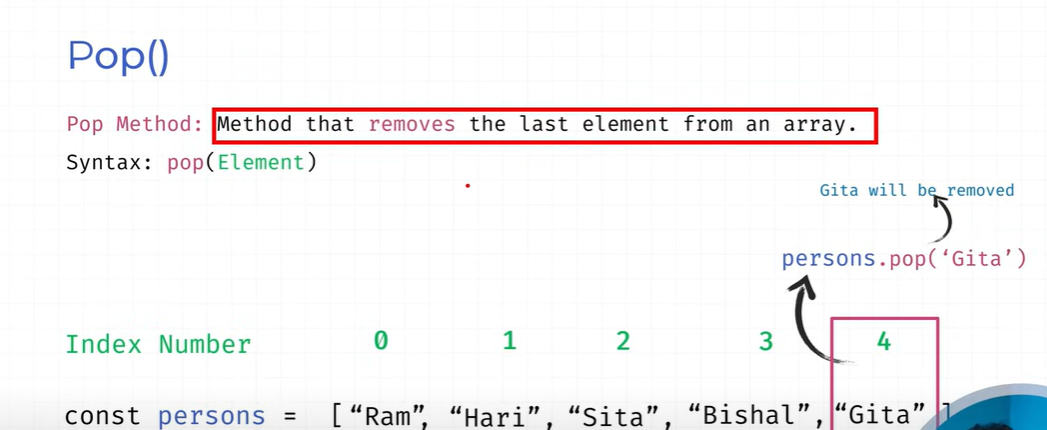
Push vs Unshift : adds the new element at the ned and beginning of the array respectively.

Push returns the new length of an array.



Pop vs Shift method removes the element from end and beginning of an array.

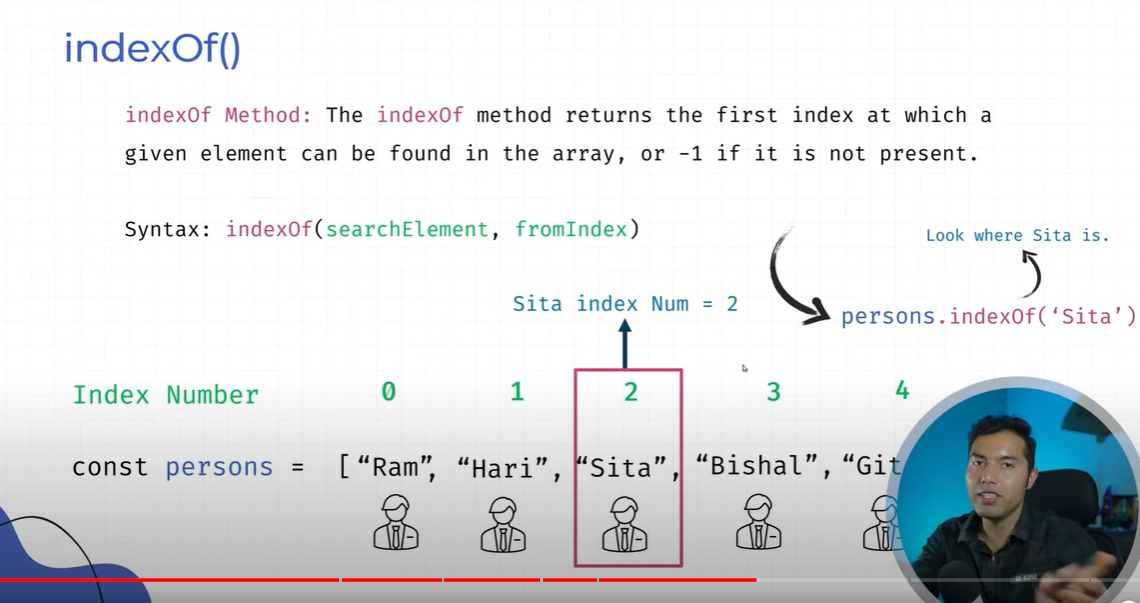
Pop methods displays the element removed form an array.

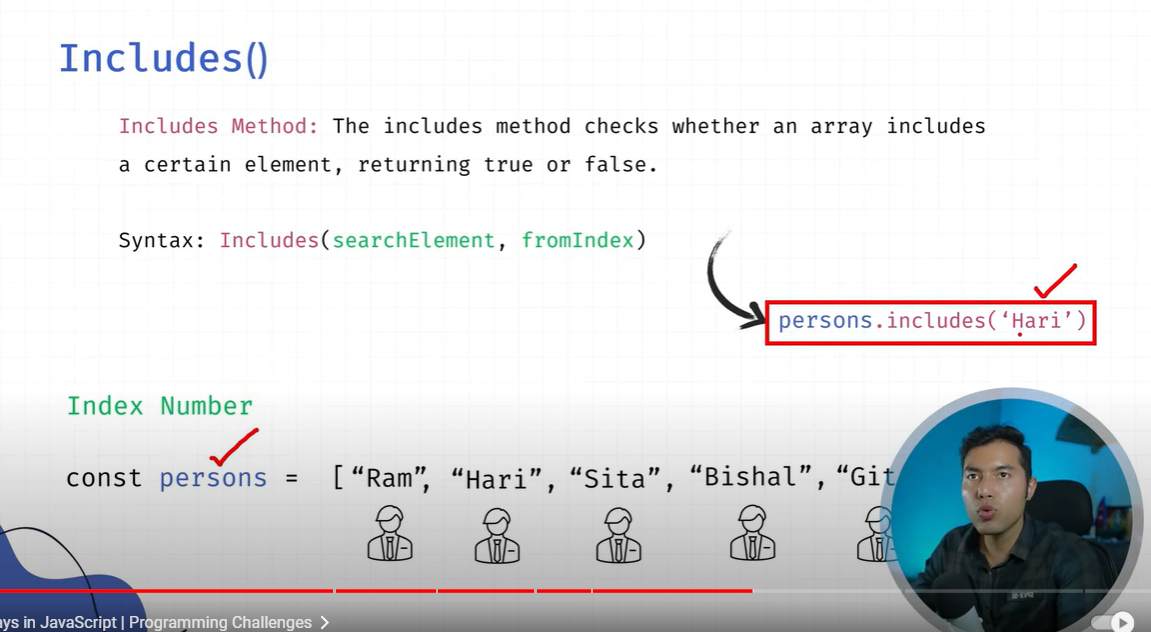


Splice : Splice method can add , remove, delete, from any position of an array

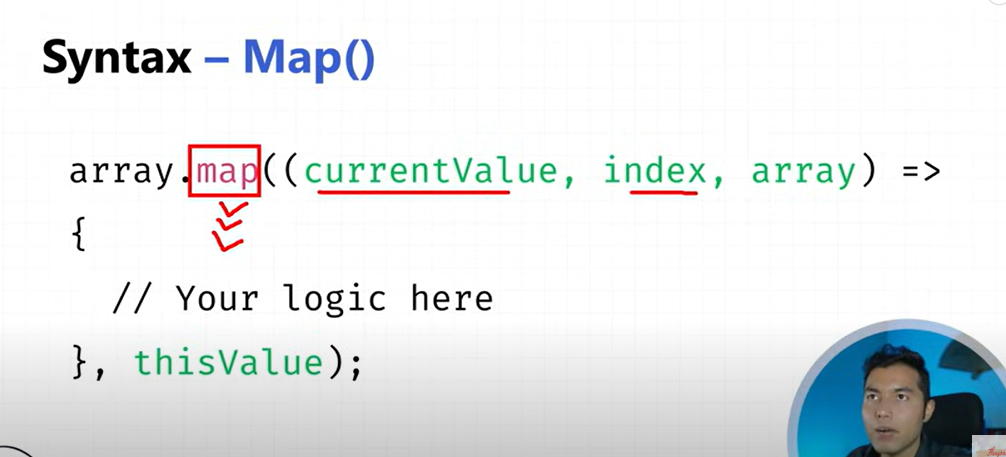
Syntax

Array.splice(start, deletecount, ‘item 1’, ’tem 2’, ………… item N);





# indexOf and lastIndexof



# **Find method**

Array.map))curElement, index, array)=>{

// logic

}.this value);

Refreet time 7 to 8

# **Filter method**

Strings

You cant use “” inside “” ; must use ‘’

You cant use ‘’ inside ‘’ : must use “”

Ex 1:

const str= "hello world"

console.log(str.length);

//output

11

ex 2:

let str1 = 'my name is \"sadhana\" & I am a \'full stack develeoper\'';

console.log(str1);

//output

my name is "sadhana" & I am a 'full stack develeoper'

let str1 = 'my name is \"sadhana\" & I am a \'full stack developer\'';

======================================

console.log(str1.length);

console.log(str1.indexOf("developer"));

console.log(str1.lastIndexOf("d"));

output

52

42

42

