

MERN Day - 7 JavaScript 2

ES6 → ECMAScript standard 6
2) Scoping 2) Arrow function 1) Variables.

var a = 10
log(a) → 10

let a = 20
log(a) → 20

let b = 30
log(b) → 30

var c = 40
a = 30
log(c) → 40
log(a) → 30

→

var a → 10

o/p: 10
20
30
40
30
10

$$c(g(c)) \longrightarrow 40$$

$$\text{var } a = 10;$$

$$c(g(a))$$

{

$$a = 20$$

$$c(g(a))$$

$$\text{var } b = 30$$

$$c(g(b))$$

$$a = 30$$

$$c(g(a))$$

$$10$$

$$20$$

$$30$$

$$30$$

$$\cancel{10} 30$$

}

clg(a)
clg(b) // undefined

Type of Scope

1) Global → any
2) local → let
 const

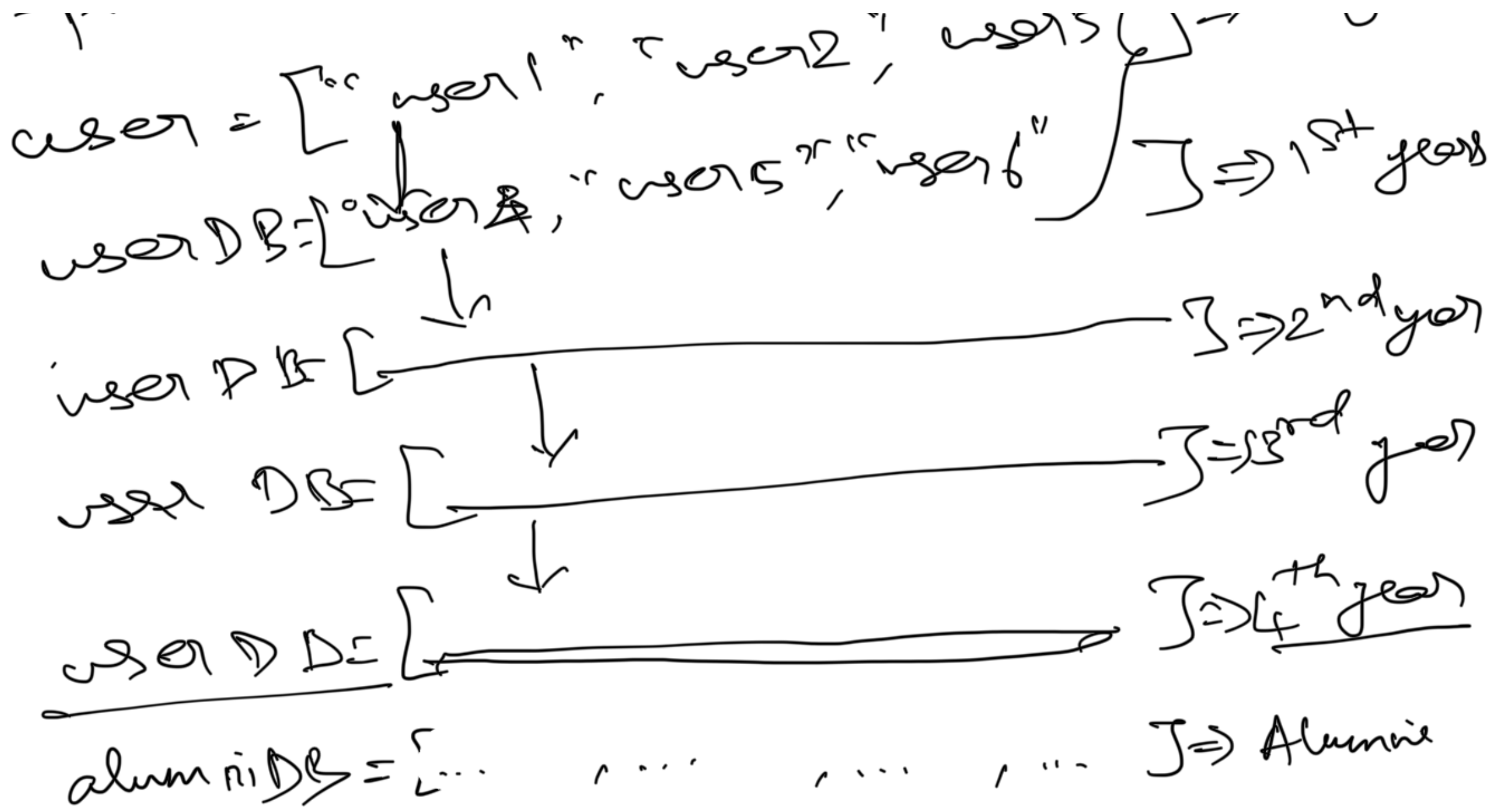
4) Ternary operator

a = 11

clg((a > 2) ? "odd" : "even")

5) Spread operator (...)

यदि → spread



Spread operator is going to spread your values in a single array.

6) Rest operator (...)

7). Destructing operator $[\quad]$
It is going to destruct the values present on the right side.

var array = [10, 20, 30, 40]

var [a, b, c] = array

c // 40

b // 30
a // 20
8). Hoisting.

Functional
Hoisting 2). variable
: undeclared

var a = 10;
a // 10

Reason: ~~Before~~ Hoisting.
a // 10

// a

// O/P: Reference

error: cannot

a before initialization.

access
let a = 10

cg(a)

Functional Hoisting

fun Hoisting() {
 console.log('');
}; fun Hoisting() {
 console.log('');
}

fun Hoisting() {

Reference Error
objects

console.log('Hello')

class class

g2

fun Hoisting() {

class func {
 constructor() {
 console.log('');
 }
}

Advanced JS.

roll back

1) callback

2) Promise

async/await

3)

X Intermediate

Promise

SS
↓
cbSSIT()

function SSIT college(message,
cbSSIT())

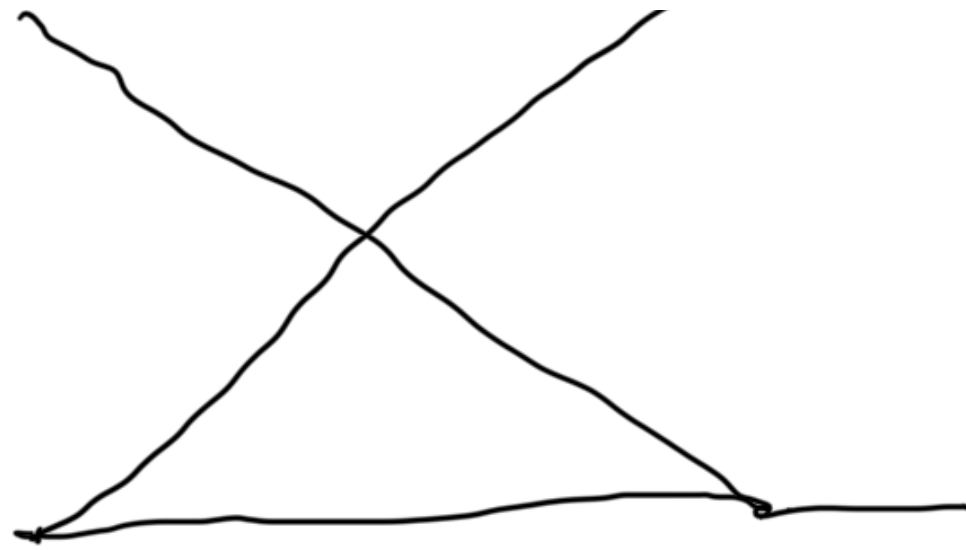
over callback It is nothing but a function

consider another
Function: callbackSSIT
function -

callbackSSIT()

function
call back SSIT & "Back to SSIT")
if 1st welcome & "Back to SSIT"
if 2nd welcome to SSIT"

SSIT college



async / await .

↳ what is async? → program
runs multiple function simulta-
neously.

DOM Manipulation

Manipulating content/page once something is loaded inside the HTML

Promise

Syntax:

new

Promise((resolve, reject) => {})