Day -11

Chai aur Javascript

→ Disable Ai
→ Because JS has many way to write same logic

Ai depend → which one has feed

Javascript

factory

Sofa . Chair . table

Tool to grab wood , so we need type of wood

Wood

- Wood
- Sagwan
- mango
- Sheesham
- walnut

we cutting teb all

Cutting

then Framing

Framing

then

Polishing

then

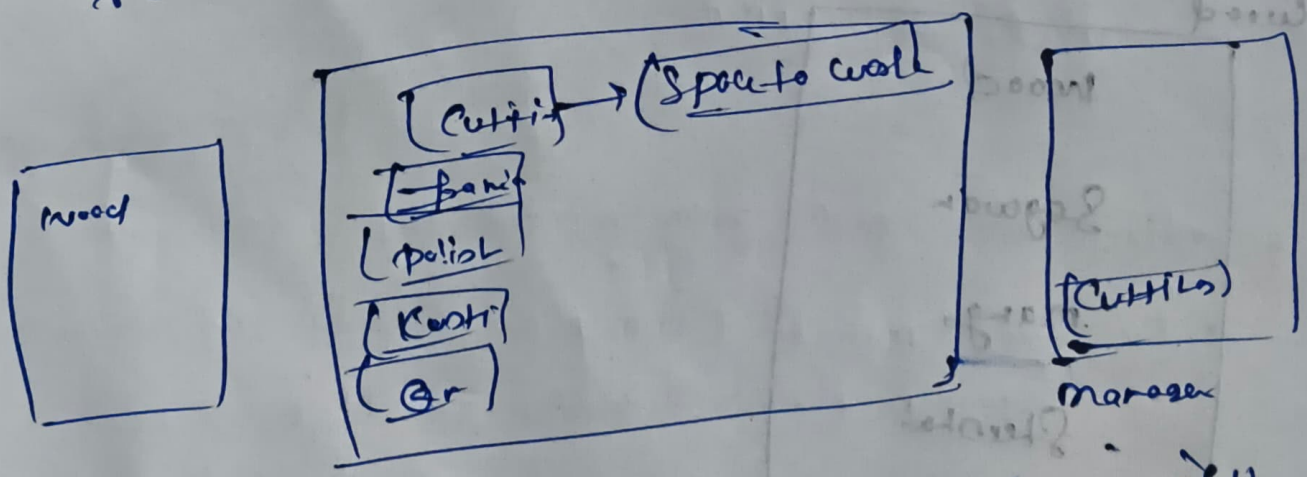
Cushion

then QA (Checking quality)

QA

and get my wood which Phase

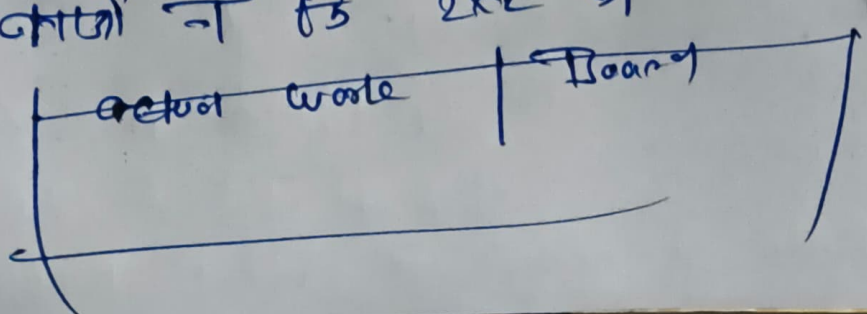
so we need manager



Space to work के लिए 5 board होने

चाहिए 6×6 का bed है तो 6×6 का

चाहिए न कि 2×2 का



पूरी factory को

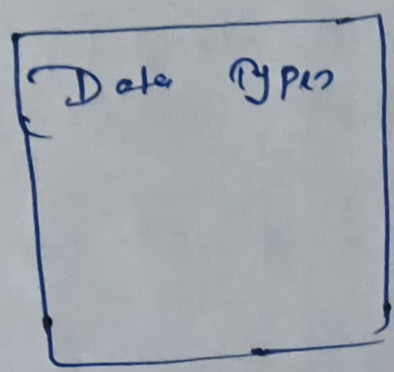
एक actual work

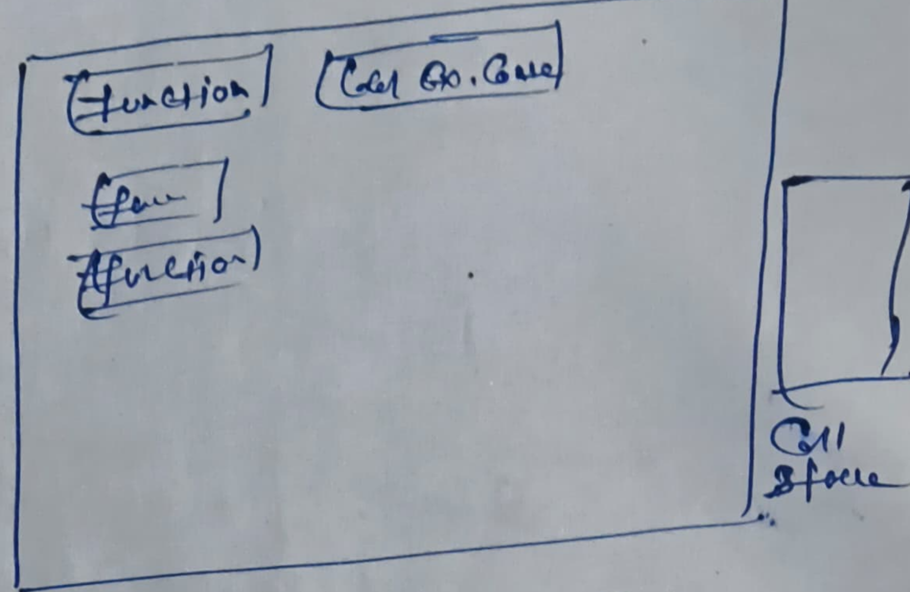
and Board एक

चाहिए

Same thing happen in JS

Global Context

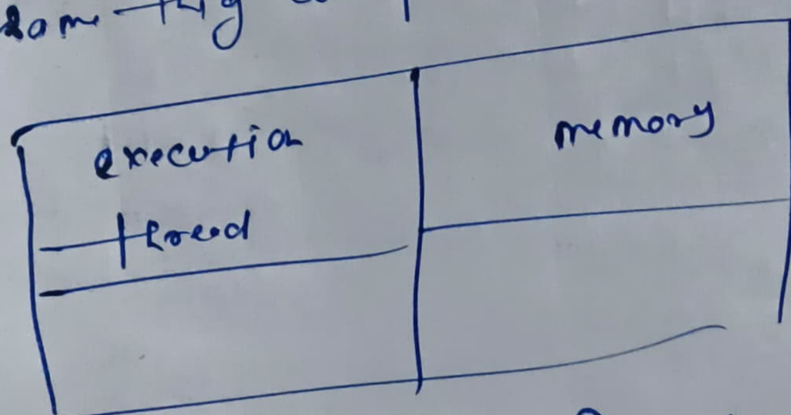




har function का अपना execution Context हो सकता है, जैसे हर mini factory का होना चाहिए।

And we also need manage and that manage Called Call Stack

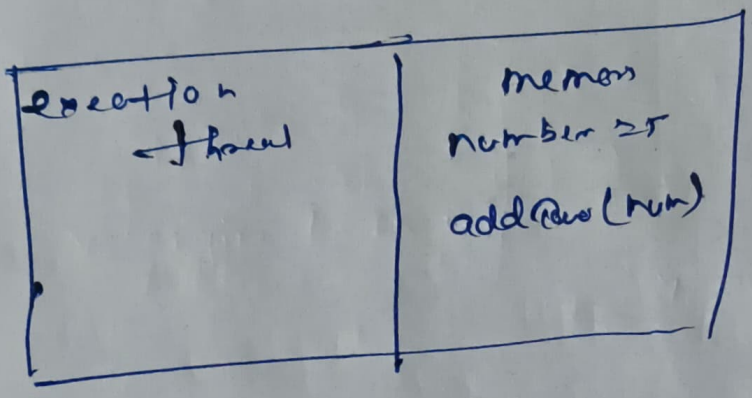
Same thing work in JS



So same thing happen for Global Context one is execution thread and other is memory

* पूरी factory को Global Execution Context कहते हैं

```
Const number = 5
function addTwo (num) {
    return num+2;
}
Const valueOne = addTwo (number)
```



Note

Programming

- क्या करना है
- कैसे करना है

पुरी Programming
- क्या करना है
- कैसे करना है

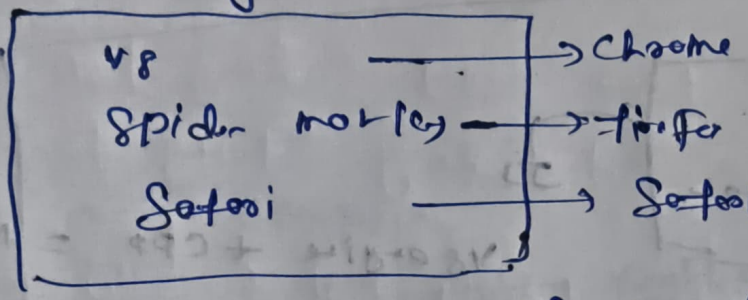
Browser

↳ understood 3 Language.

↳ HTML
↳ ② CSS
↳ ③ JS (JavaScript)

↳ JS को Run के लिए भी Parser या engine होता

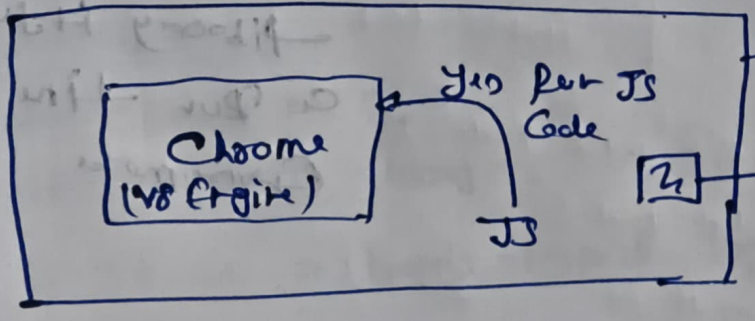
JS engin name



अगर ये engine नहीं है किसी भी तरह JS Run नहीं हो सकता है,

JS engine to Run JS Code

C, C++ Compiler ⇌ CPP Compiler
Java = Java Compiler
JavaScript = JS engine



Can i Run JS outside browser → No

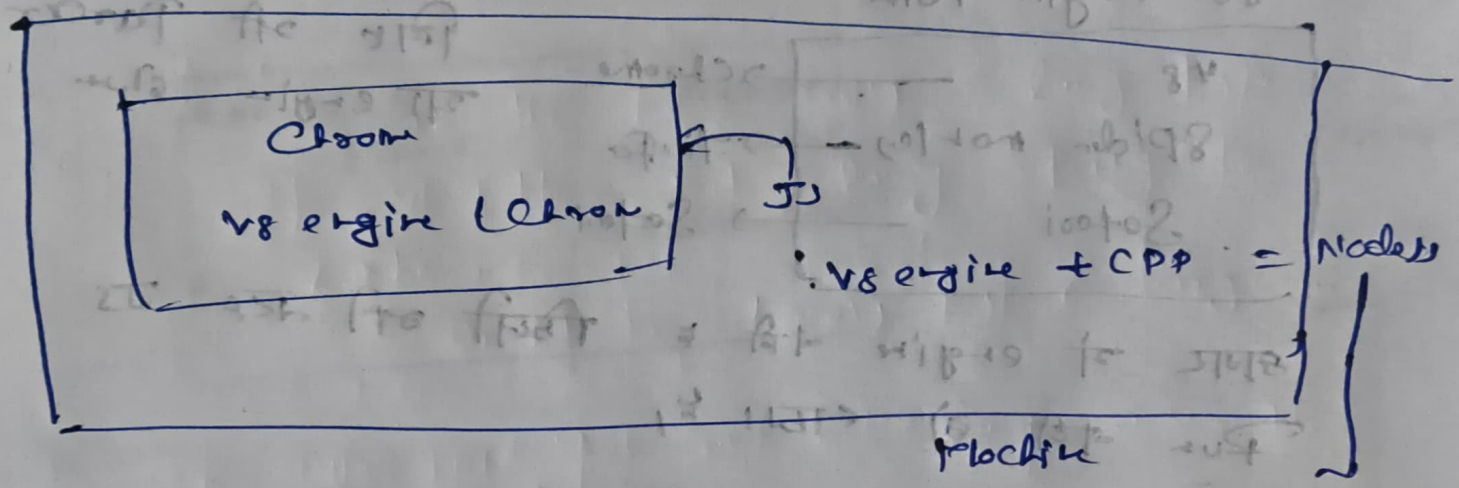
V8 engine + CPP = Nodejs

↳ when i just put in Laptop

~~the~~ ~~that~~ Can Run JS

means now i Can Run JS outside

~~of~~ ~~the~~ browser



Node JS.

↳ JS is ~~framework~~ / library → No, it is neither ~~framework~~ / library it just a Run time Environment

Console → यह is just print output of JS.

Sources → कौन सा file कहा से load हो रहा है,

network → कहा से क्या Request / Response आ रहा है

(A) Internal JS

```
<html>

<script>
    function sayHello(){
        Console.log('Hellow ji')
    }
</script>
</html>
```

not secure,

External JS

Script.JS

Console.log('I am a code in Sc.JS file')

Script.JS

var fname = 'Piyush';

var lname = "Garg";

// JS is a loosely typed language

Console.log('value of fname is', fname)

```
frame = 'Hello';
Console.log('value of frame is', frame) // Hello
frame = 32
Console.log('value of frame is', frame); 32
```

↳ So in this frame easily Change type of variable, just Change String to number.

↳ So it is loosely typed language.

* In TypeScript you can't Change a variable data type.

Browser not understand TS typescript that Browser Compile TS to JS and then browser will be understand.

* function is a set of instruction

```
function addNumbers(num1, num2){
  var result = num1 + num2;
  Console.log('Result is', result)
}
addNumbers(2,3)
addNumbers(3,4)
```

Conditionals

↳ Sometime we want code Run on a particular Condition

for age 18 or 18+

Console.log ('vote is allowed')

Console.log ('you are not allowed to vote')

↳ to below 18

this both need to Run under a Condition

// Condition = ya toh haga ya fir nahi

// Condition = True or false

```
let age = 19;
Console.log (age > 18); // True
```

```
let age = 18;
Console.log (age > 18); // False
```

```
let age = 18;
Console.log (age ≥ 18); True
```

Note Condition always Return True / False (Boolean) return

```
var age = 22;
var Condition = age >= 18;
var Condith
    if (Condition) {
      Console.log('Vote is allowed')
    }
```

Verbose
1. Necessary
2. too much

```
var age = 12;
var Condition = age >= 18;
    if (Condition) {
      Condition ("Vote is not allowed")
    }
```

```
var age = 12;
var Condition = age >= 18;
var Condition
    if (Condition) {
      Console.log('Vote is allowed')
    } else {
      Console.log('Vote not allowed')
    }
```

- but what for we have multiple condition
- then this code not for that
so we need something new

Categories

0 - 12 — child
13 - 18 → teen
20 — 40 — adult
40 + → senior

```
var age = 22;
var childCondition = age >= 12;
var teenCondition = age <= 19;
var adultCondition = age <= 40;
var seniorCondition = age > 40;

if (childCondition) {
    console.log('you are a child');
}

if (teenCondition) {
    console.log('you are a teen')
}

if (adultCondition) {
    console.log('you are a adult')
}

if (seniorCondition) {
    console.log('you are a senior')
}
```

Scripts

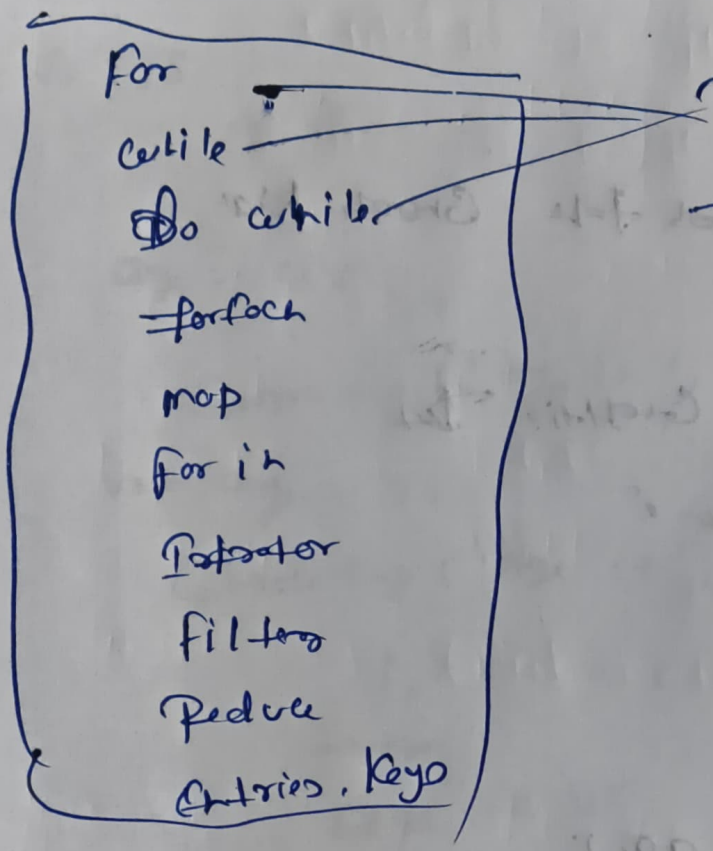
var teenCondition = age > 12 && age <= 19;

var adultCondition = age >= 20 && age <= 40;

```
if (childCondition) {
  console.log ('you are a child')
} else if (teenCondition) {
  console.log ('you are a teen')
} else if (adultCondition) {
  console.log ('adultCondition')
} else if (seniorCondition) {
  console.log ('you are a senior
} else {
  console.log ('seni Condition hi
  false ho gyi...');
}
```

Cater Condition true at some point that stop remaining code (not execute remaining

Loops




```
for (initializer; condition; increment) {
    // Code block to be executed
}
```

```
for(var x = 1; x <= 10; x++){
  console.log('Here is the value', x);
}
```

for

- DRY Rule

x	condition = x <= 10	x = x+1	Output
1	True	(1 <= 3)	1
2	True	(2 <= 3)	2
		(3 <= True)	3

For →

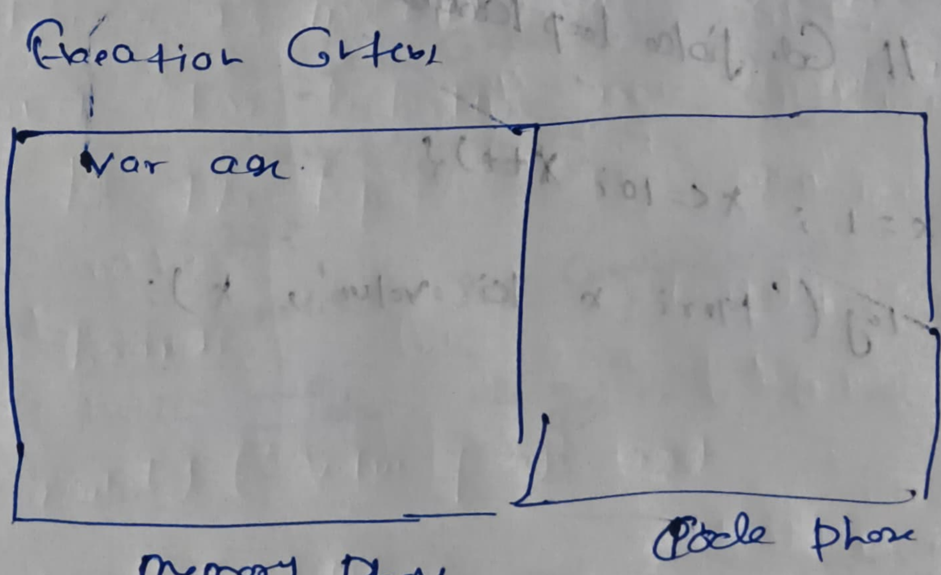
While → Code ko bar bar Execute krna hai

? why you know the Execution Context

Console log then?

var age = 30;

Console.log('Age is', age);



Hoisting

A JS

```
age = 45;
   Console.log('Age is ', age);
var age = 32;
   Console.log('Age is', age);
```

↓

Age is 45
Age is 32

Note ÷ debug point
↳ जिसपे लगाते है वह line अभी execute नही हुआ है तब तक का execution Context देख पाते है

var x = 23

↳ Execution Code में x = undefined

memory Phase में x = 32 रहती है,

Script.js

~~hello()~~;

Console.log('Age is ', age);

~~function~~ hello (){

Console.log(' This is hello')

}

Hoisting

→ It is a JS mechanism where variables function and class declaration are conceptually move to the top.

~~(B)~~ brainus

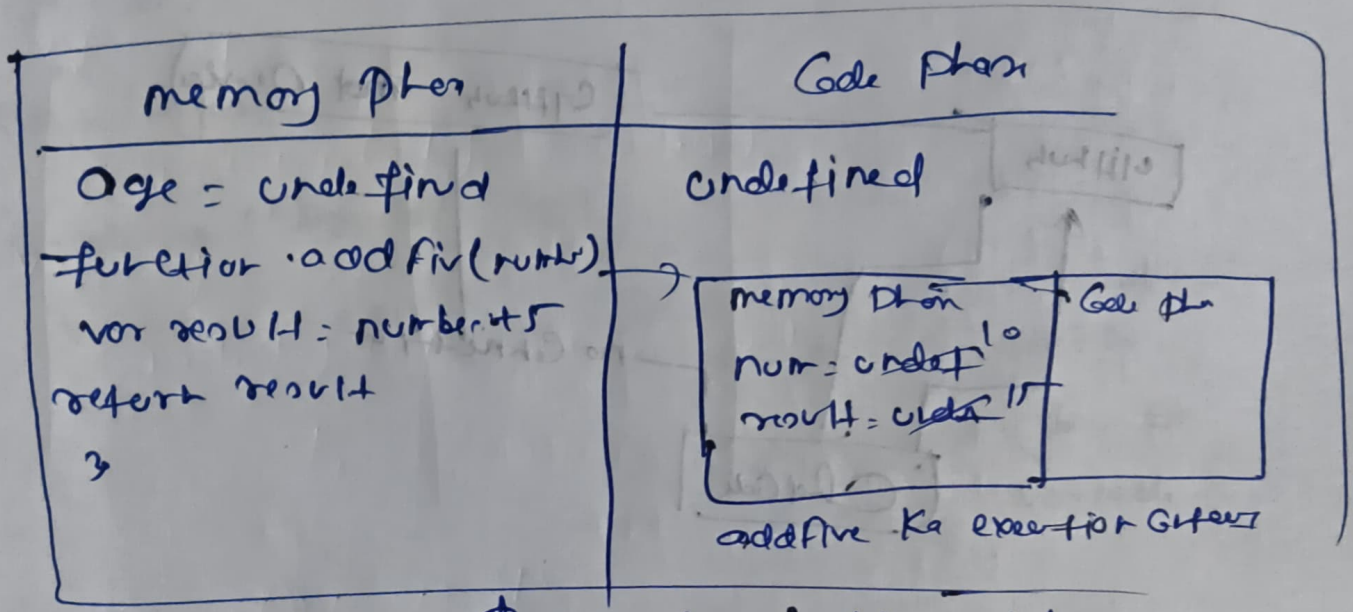
Console.log (' value of age', age)

var age = 45

~~addFive (10)~~ Console (' Adding 5 to 10 :', addFive(10));

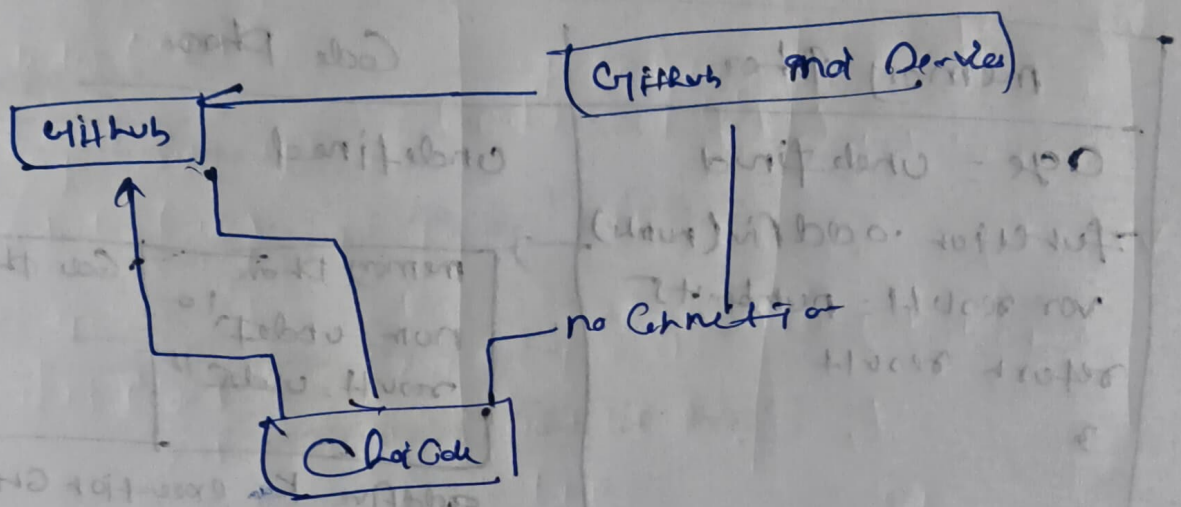
Console.log((' value of age is ', age)

function addFive(number){
var result = number + 5;
return result
}



JS engine Top to bottom जाता है
फिर और memory Phase में data लेता है
Code Part execute करता है
इसलिए variable पहले undefined and then
value put हो जाता है

JS हमेशा अकेले में सीखो



~~GitHub Repo~~

Git

GitHub Classroom