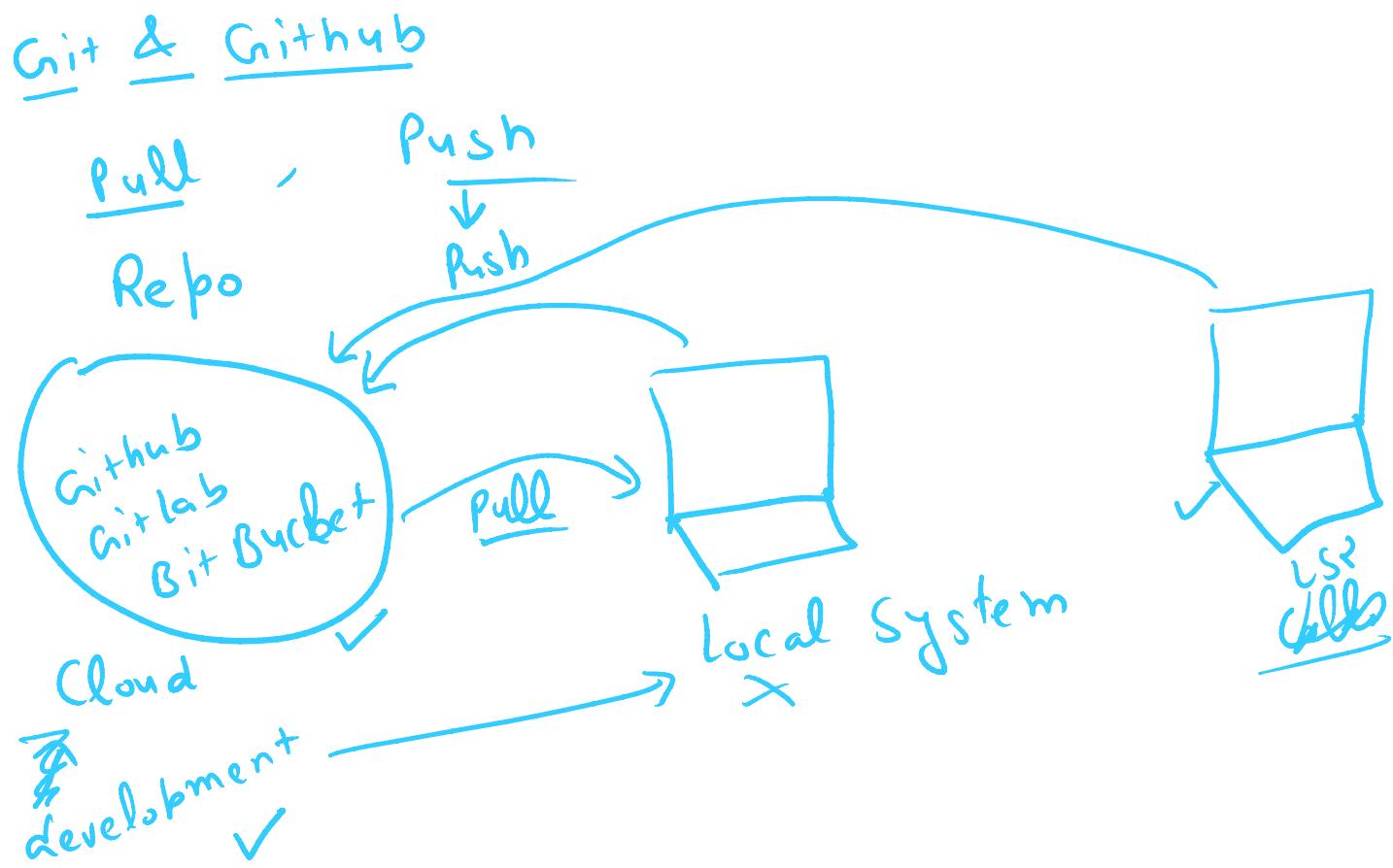


Previous Class

24 August 2022 19:45

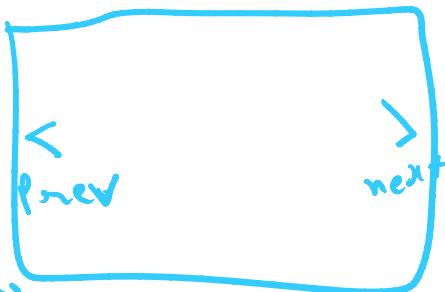


What

Why we use JavaScript

If you want to change the content of your website without reloading the webpage.

dynamic



Desktop Applications

image-1 → image-2

Programming language
Fully Integrable
HTML CSS

do

It can change
the content in Browser
only.

can't do

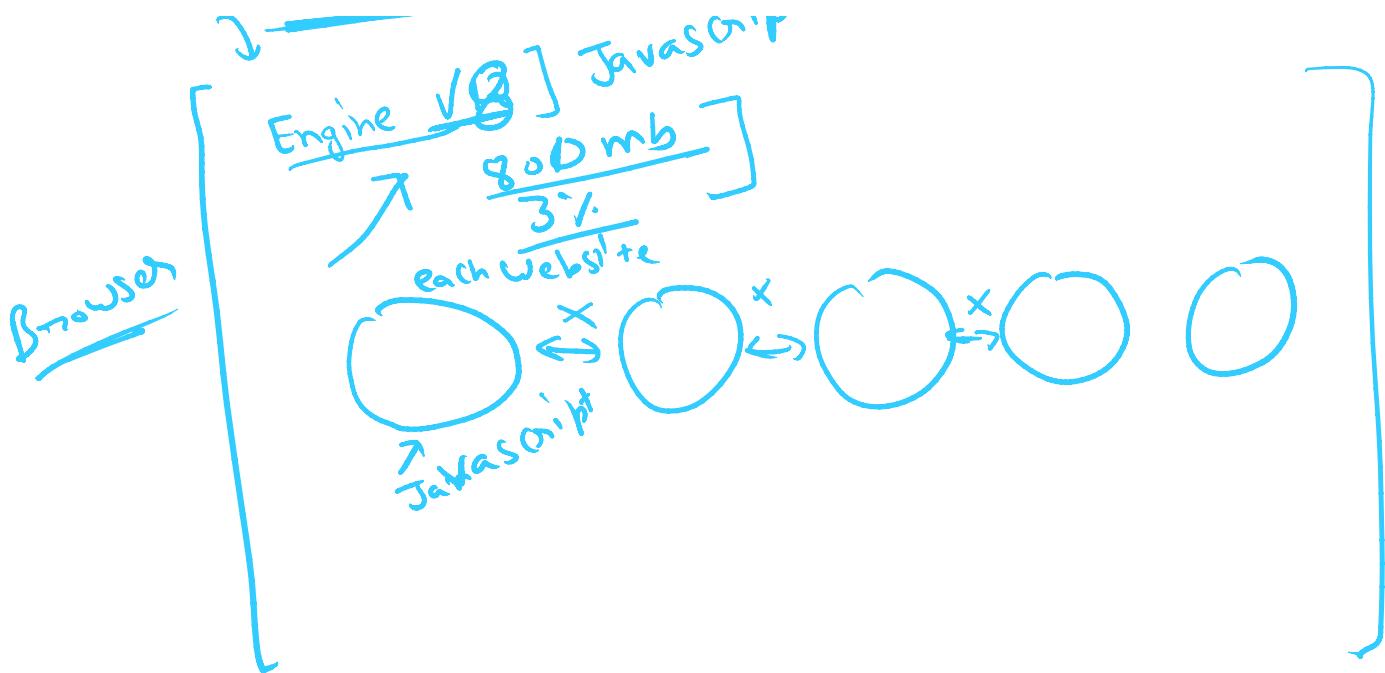
It can't perform any
action on our hard
disk.

Javascript code of
a website is restricted
to single tab

1nB 5%

Ram some Space
[... 10] Javascript ✓

7



Installation Of Nodejs And VScode

24 August 2022 20:11

1. Open the browser and search for nodejs
2. Click on Download or go to <https://nodejs.org/en/download/>
3. Download the software as per you OS configuration
4. Click on next, accept T&C and next
5. Don't change location
6. Don't check to install chocolatey
7. Finish the installation

Verify Nodejs Installation:

1. Open CMD
2. Type **node -v**

Version of your node server should be displayed

How to open a folder in Vscode

1. Click on file and then click on open folder

To open the terminal

1. Click on terminal then click on new terminal

Print and comments

24 August 2022 20:11

Printing in javascript

```
24-08-2022 > JS how-to-print.js
1   console.log('Hello World!')    ↪ arguments
2   console.log("hello","my name is bhavesh") ↪ single           ↪ multiple
      ↪ statements
```

```
bhave@DESKTOP-7NME503 MINGW64 ~/Desktop/august-2022/24-08-2022 (main)
$ node how-to-print.js
Hello World!
hello my name is bhavesh
Space
bhave@DESKTOP-7NME503 MINGW64 ~/Desktop/august-2022/24-08-2022 (main)
```

Comments

```
1 // console.log('Hello World!') --> single line comment
2 /* (console.log("hello","my name is bhavesh"))
3   console.log("hello","my name is bhavesh") */ ↪ multiline comment
```

operators

24 August 2022 20:12

Data types

24 August 2022 20:12

Null vs undefined

letters
↓
houses
any

memory

← [Bhavesh
Sarabha]

undefined

Sarabha
↓
Space hole X
empty
null
memory is empty

Variables

24 August 2022 20:12

Variables

25 August 2022 20:08

Variables

25 August 2022 20:08

[loosely typed language]

functions

a variable

In Java Script

In Java

int

float

String

char

data types

let

var

int a = 0

int a = "bhavesh"

to declare are:

1. let
 2. var
 3. const

harmful
for software

 very bugs

many bugs
Security issues

only variables
not defining any
data type

good → developer

Let y
↓
any value

functions

classification

let

&

functions

const

↳ reassignment

The diagram illustrates variable shadowing in JavaScript across three scopes:

- Global Scope (Left):** Contains the code `const a = 3;` and the value `a = 4 X`.
- Function Scope (Middle):** Contains the code `var a = 3;` and the value `a = 4 ✓`. This scope shadows the global variable `a`.
- Block Scope (Right):** Contains the code `let a = 3;` and the value `a = 4 ✓`. This scope shadows both the function variable `a` and the global variable `a`.

A vertical line with arrows at the bottom indicates the scope boundaries, showing how variable declarations affect the visibility of variables in nested scopes.

Diagram illustrating the execution flow of the code:

- Execution starts at **undefined**.
- An arrow labeled **Dec + Assign** points down to the statement **let a = 2**.
- An arrow labeled **declaration** points up from **let a = 2** to the declaration **let a**.
- An arrow labeled **assignment** points from **let a** to the assignment **a = 4**.
- An arrow labeled **Assignment** points from **a = 4** back down to the value **a = 2**.

let x ✓
let x ✗ ✓ on the

let x ✓
let X ✓ on the
JavaScript is a case-sensitive language

Naming rules

- Names of variables

 1. Starting with letter
 2. Starting with - (underline)
 3. Starting with \$ (Dollar)

3. Starting with T

numbers or other special characters

Keywords can't be used as variables

Numbers, strings, boolean, null, undefined

Primitive data types

↓
immutables, can't replace
with any other character

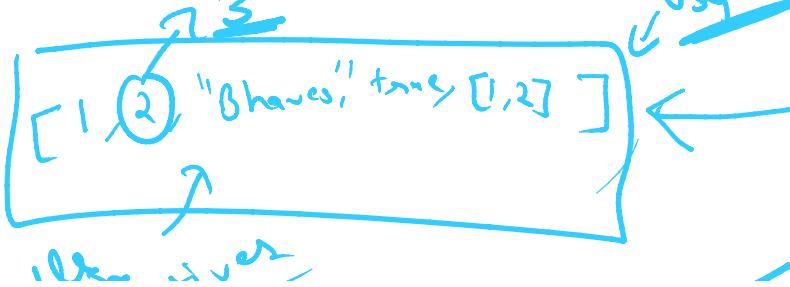
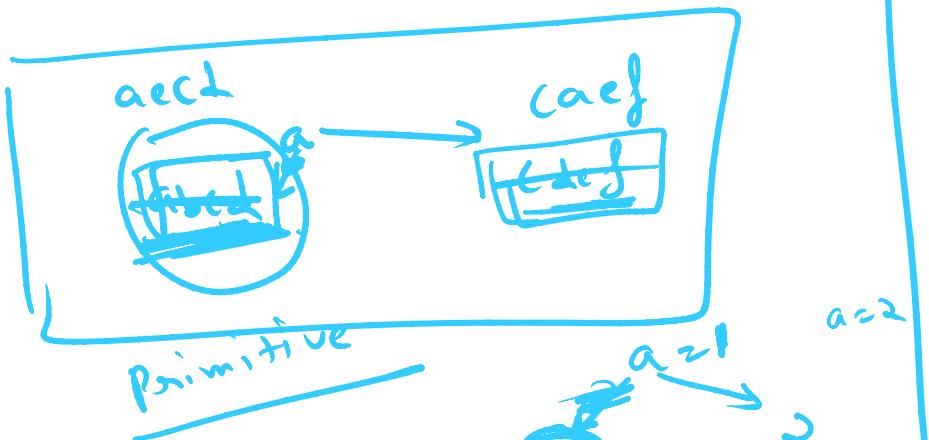
let a = "abcd"
 ^
 0 1 2 3

String is immutable

a[1] = e

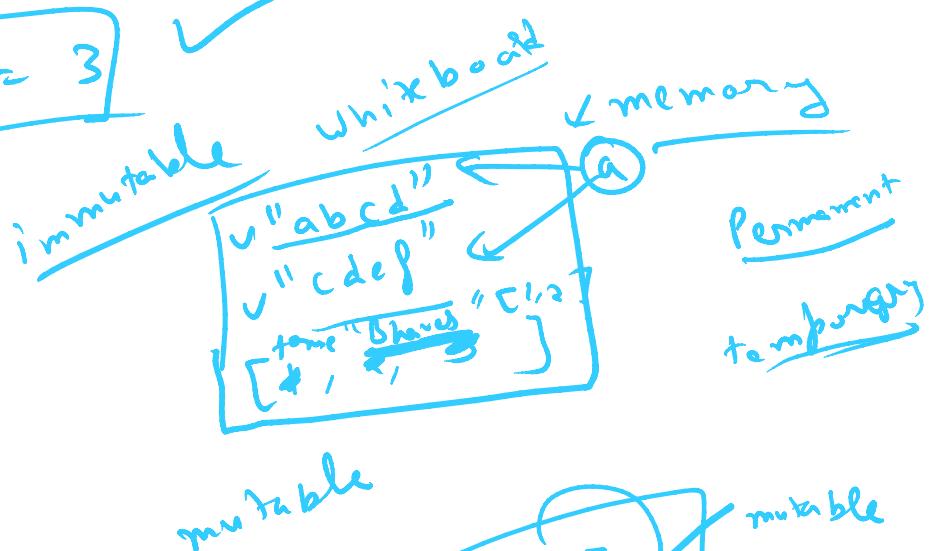
let a = ("abcd")
a = ("cdef")

[non-primitive]
Array
Objects

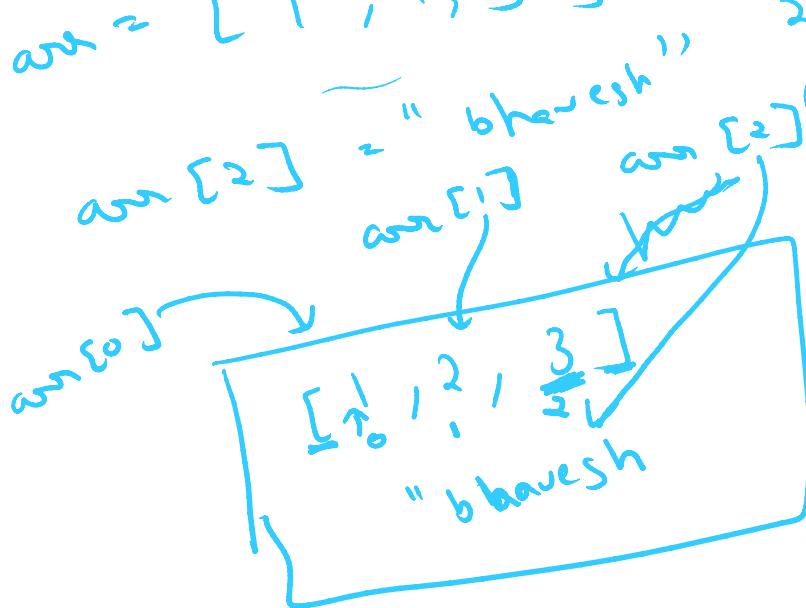


#strict
av
typescript

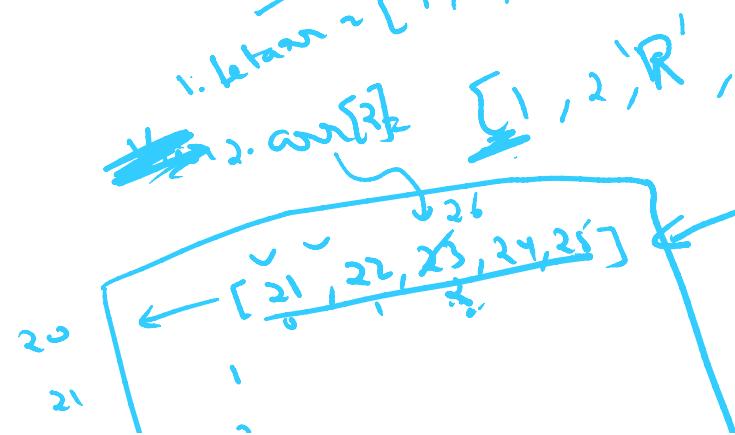
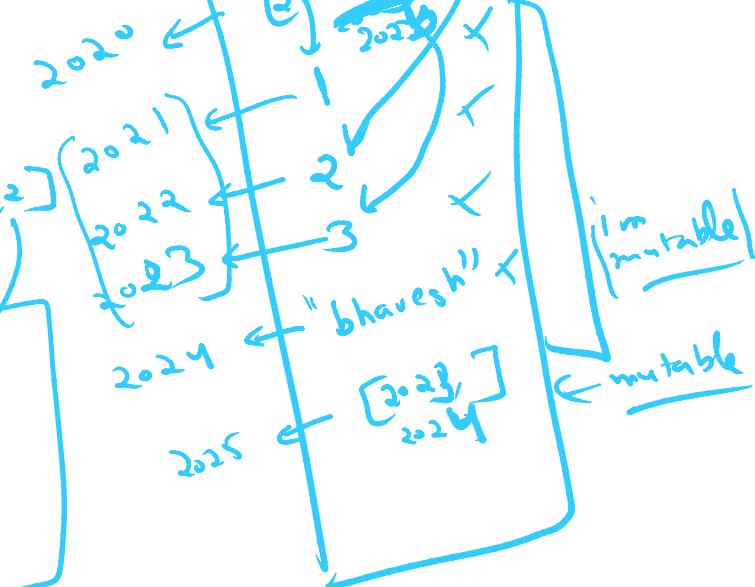
~~mutable~~
non-primitive
 $\text{arr}[1] = 3$



$\text{arr} = [1, 2, 3]$



mutable



{
 ?
 new Array

1. let arr = [1, 2, 3, 4, 5] ✓

2. arr[1] = 26 ✓

21

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

23

24

25

26

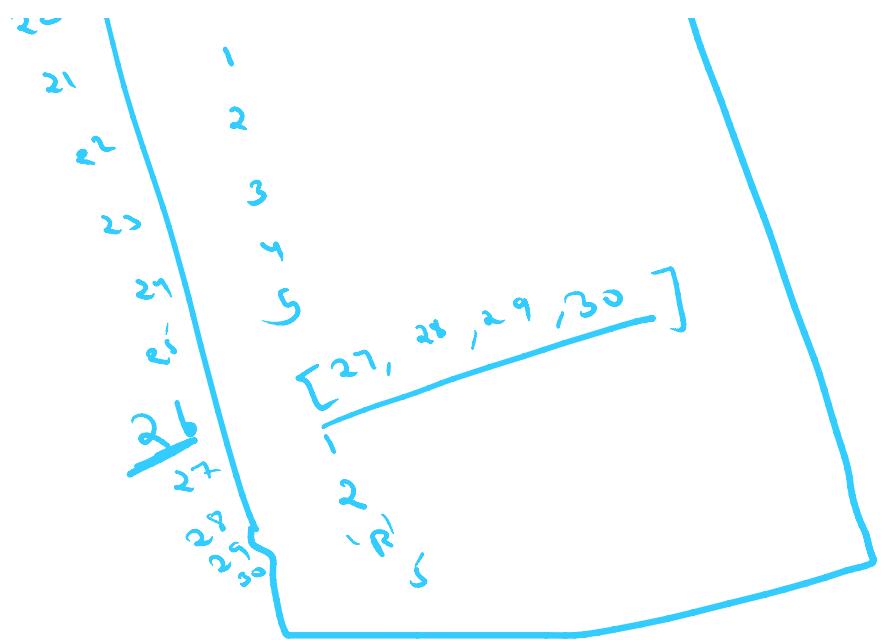
23

24

25

26

23



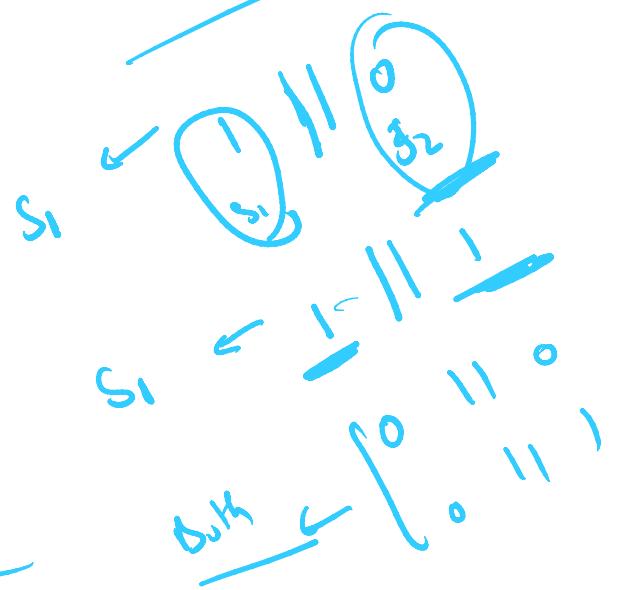
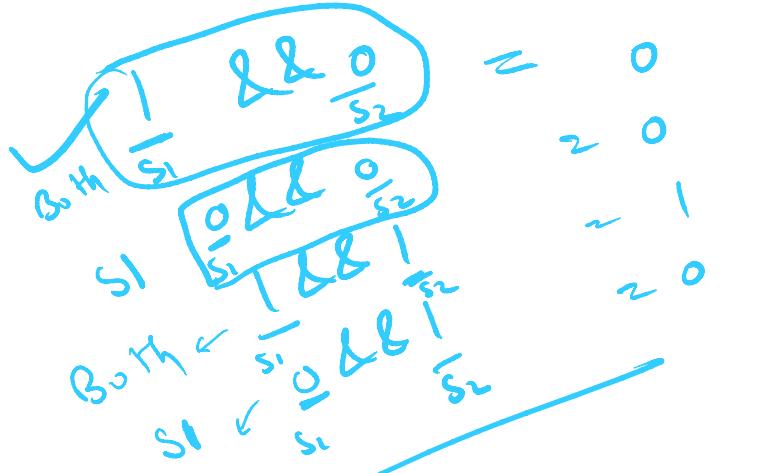
operators

25 August 2022 20:08

$\&&$ → and
 $\|$ → or

!

$!0 = \text{true}$
 $! \text{false} = \text{false}$



$==$ → true
 $==$ → false
 $==$ → true
 $==$ → false

values are same
Data type is same

$!=$
 $!=$
 $>$
 $>$
 $<$

$\sqrt{c_i}$

Let vs Var

26 August 2022 20:20

let is not redeclarable

let a = 2



error

let a = 5

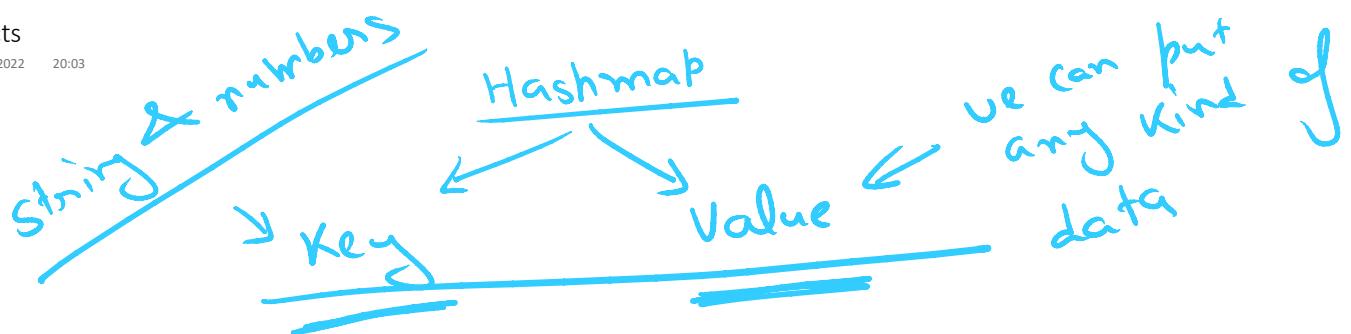
Var is redeclarable

Var a = 2

Var a = 5



no error



Let obj = { "name": "bhavesh",
"experience": 3 }

Let arr = ["bhavesh", 3, "abc"]

let obj = { 0: "bhavesh",
1: 3,
2: "abc" }

{ arr[0] → obj[0]
arr[1] → obj[1]
arr[2] → obj[2] }

= Array in Javascript is nothing
object with all ^{continuous} numerical

// ...
but an object with all numerical
keys.

...
let obj = {
 o: Obj2,
 experience: 3
}

Obj2 = {
 name: "bhavesh"
}
3

Object.keys(obj) → [o, 'experience']

Object.values(obj) → [Obj2, 3]

{
 name: "bhavesh"
}

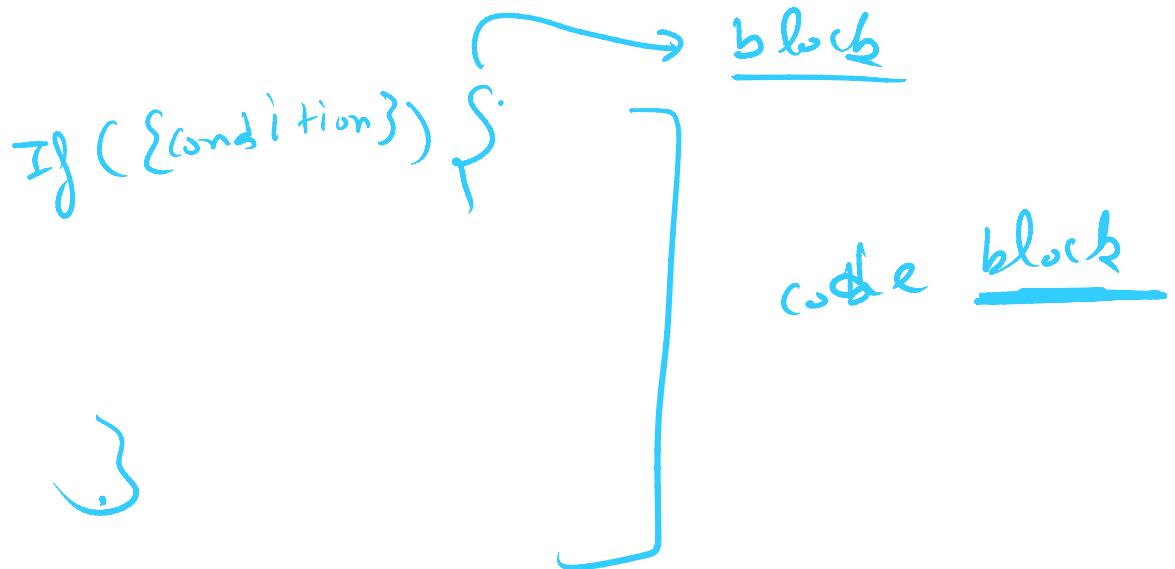
obj[o] → Obj2 →

Object.keys(obj[o]) → [name]

Object.values(obj[o]) → ["bhavesh"]

If Else

26 August 2022 20:03



Code examples:

```
If ({condition}) { } else { }  
If ({cond.}) { } else If ({cond.}) { }
```

Ternary Statement

Truthy values	
false	negative
0	none
empty string $\leftarrow ("")\right.$	"a"
false	true
null	[]
undefined	{ }

... ? {S1} : {S2}

{ condition } ? { s1 } : { s2 }

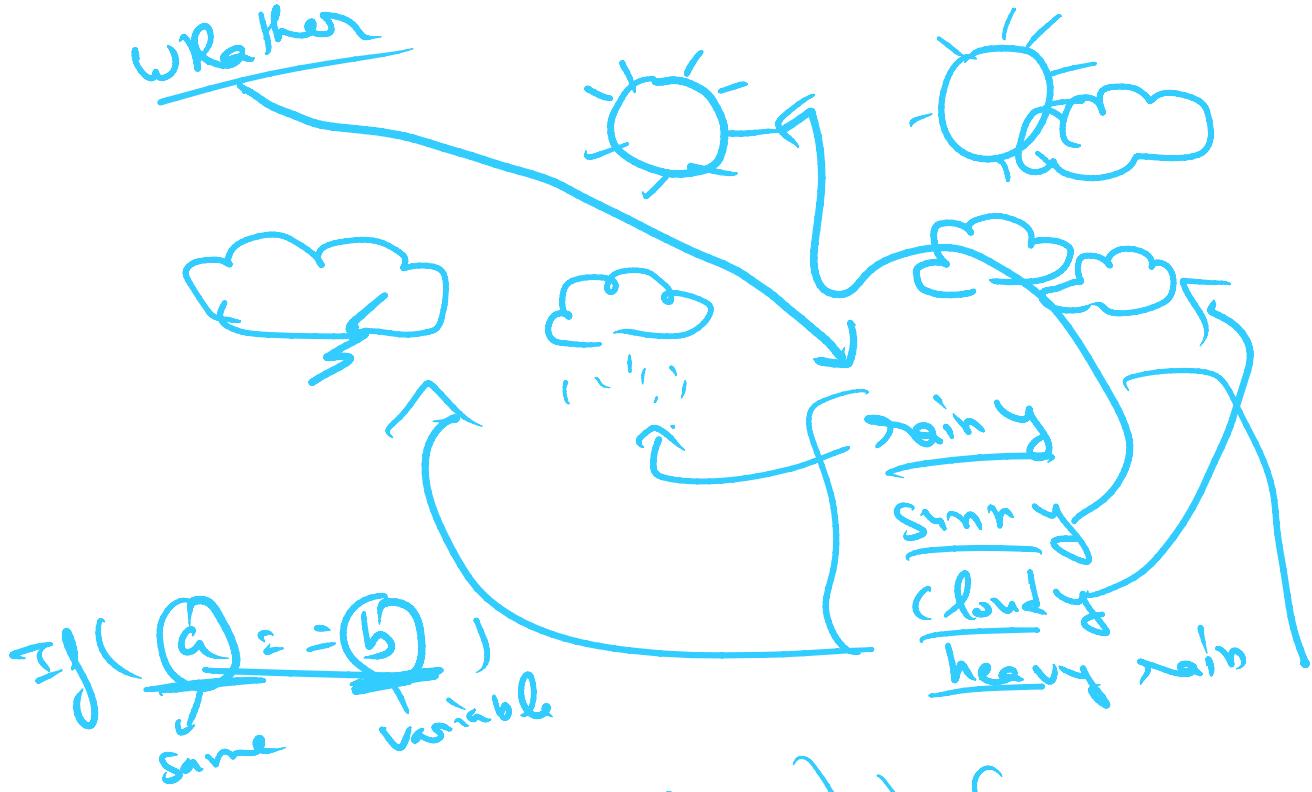
If (condition) {
 s1
}, else {
 s2
}

Switch

27 August 2022 20:21

When there are multiple equal to conditions and only one is going to be true

~~weather~~



Switch (weather) {

 case ("rainy"): {

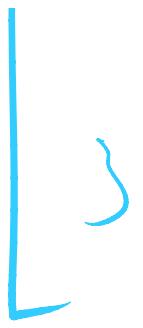
 console.log ("It's raining");

}

 case ("sunny"): {

 console.log ("It's a sunny day");

}

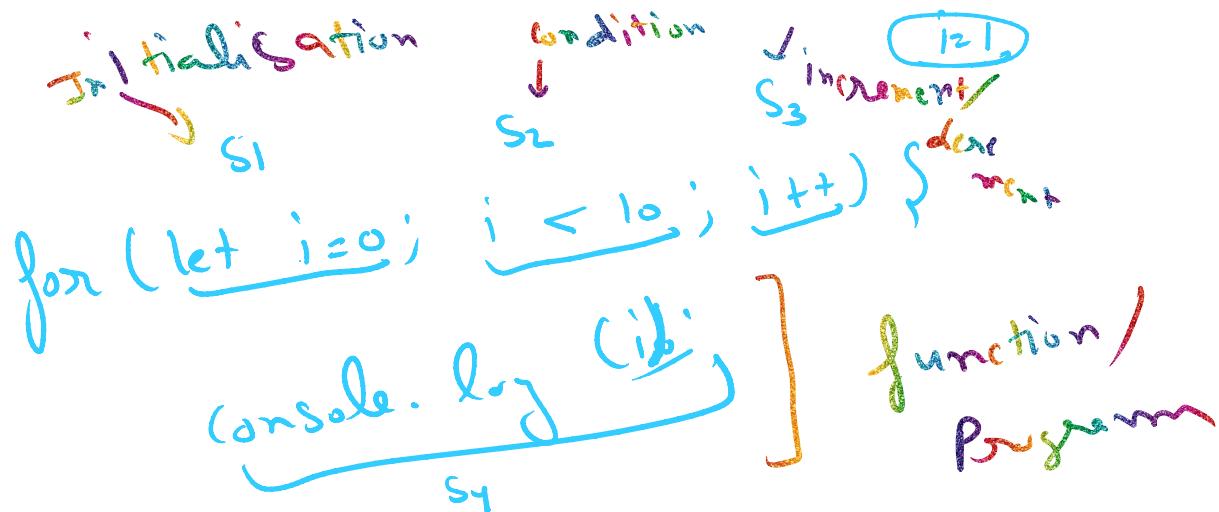


Switch

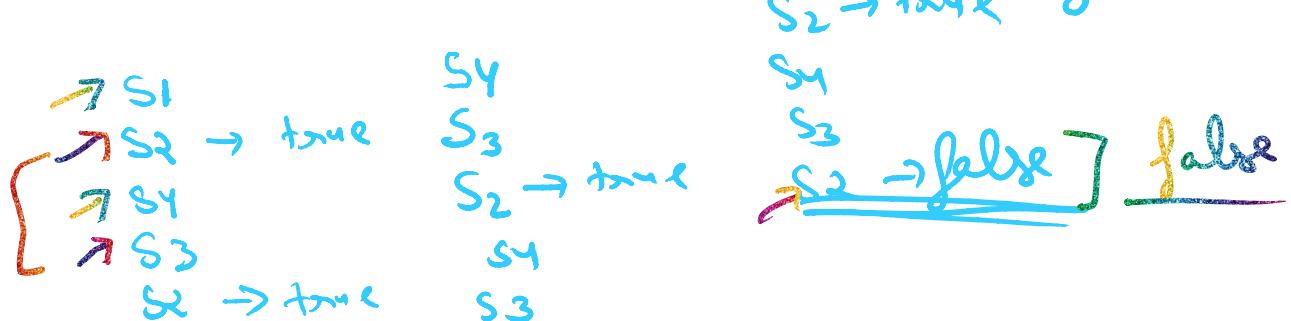
For loop

27 August 2022 20:41

Break and continue



}



$i = 0$
 $j = 0$

3

```
let arr = [[1,2,3],[4,5,6],[7,8,9]]
for(let i = 0; i < arr.length; i++) {
  for(let j = 0; j < arr[i].length; j++)
    console.log(arr[i,j])
}
```

arr[0][0]

$arr[0].length \approx 3$
 $[1,2,3].length \approx 3$

while

27 August 2022 20:41

S_1
While(condition){
} $S_2 \rightarrow$ program

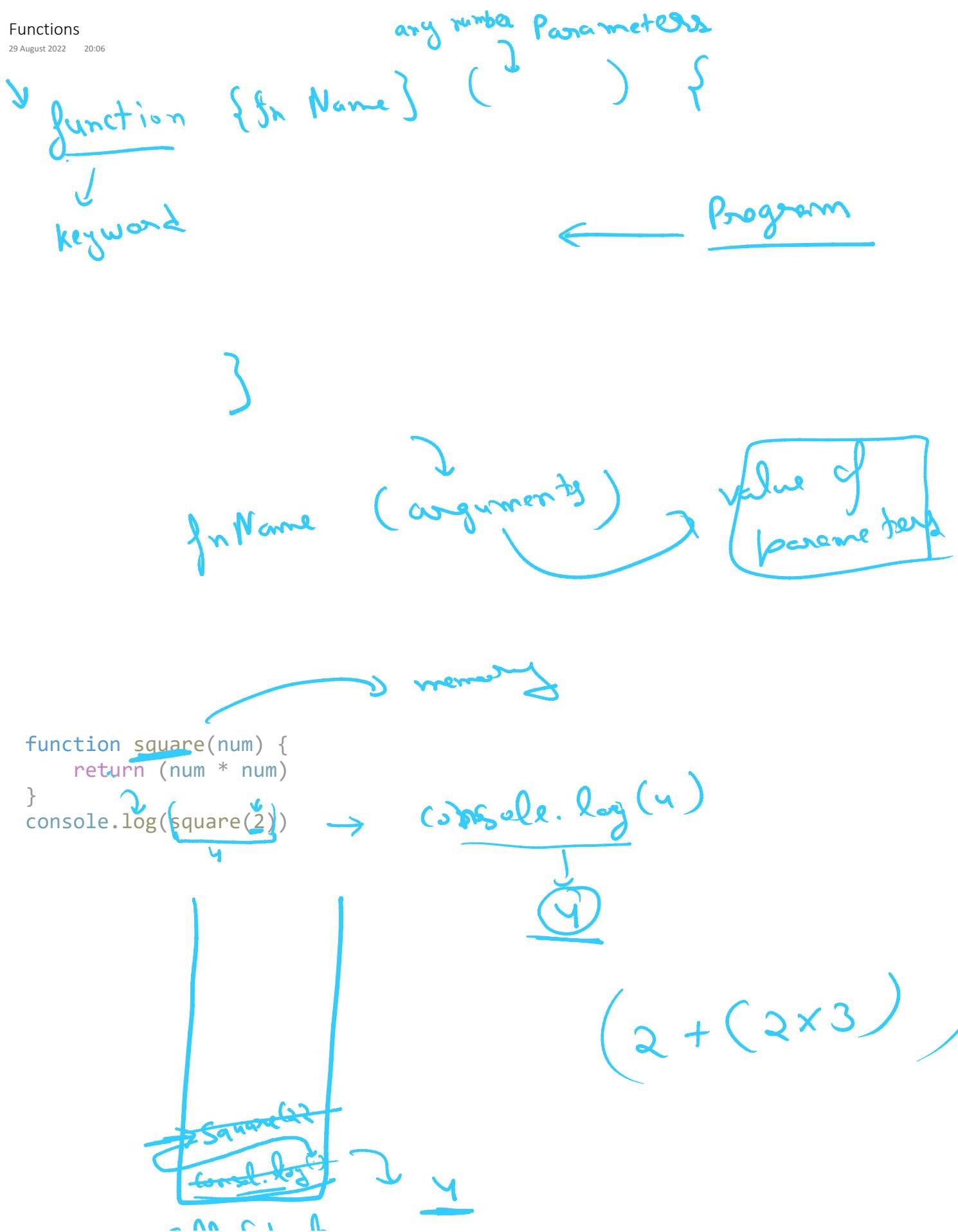
$S_1 \rightarrow$ true
 S_2
 $S_1 \rightarrow$ true
 S_2
 $S_1 \rightarrow$ false

do {
 S_2
} while (condition)

S_2
 $S_1 \rightarrow$ true
 S_2
 $S_1 \rightarrow$ true
 S_2
 $S_1 \rightarrow$ false

Functions

29 August 2022 20:06



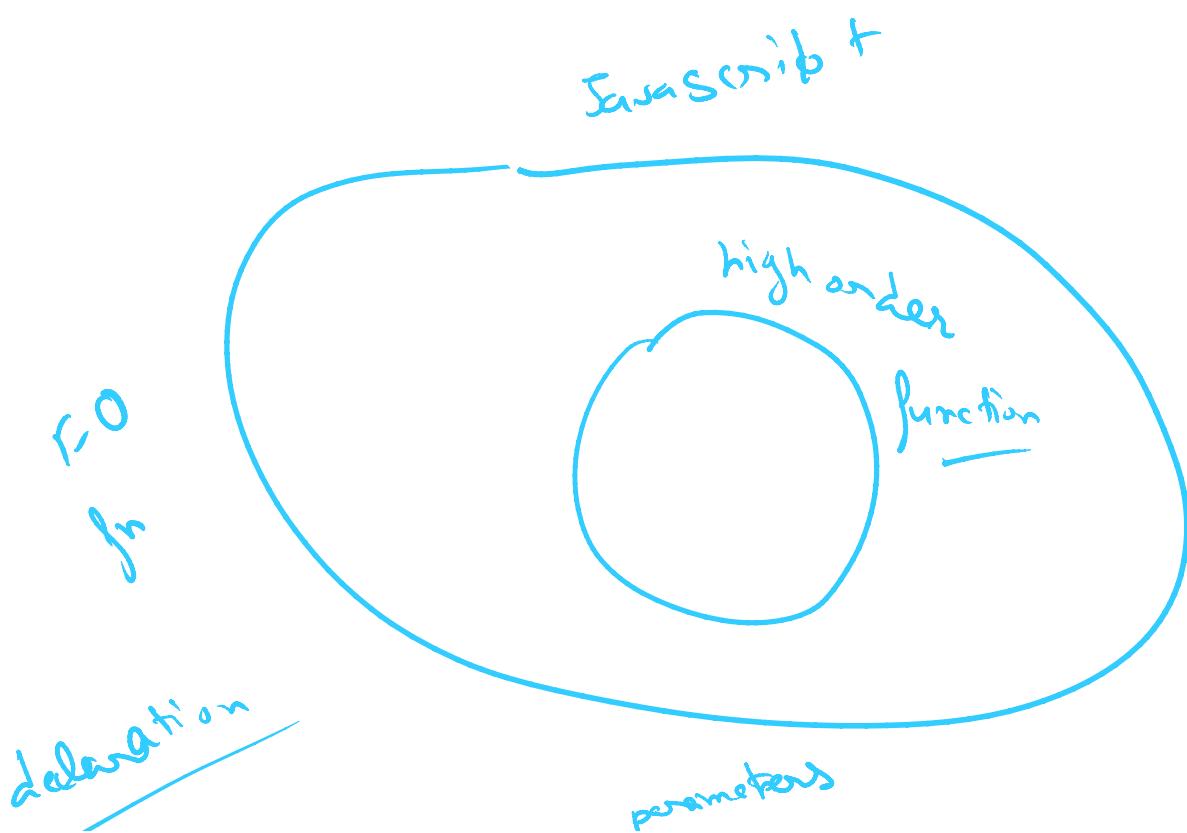


```

function square(num) {
    console.log(num * num)
}
square(2)

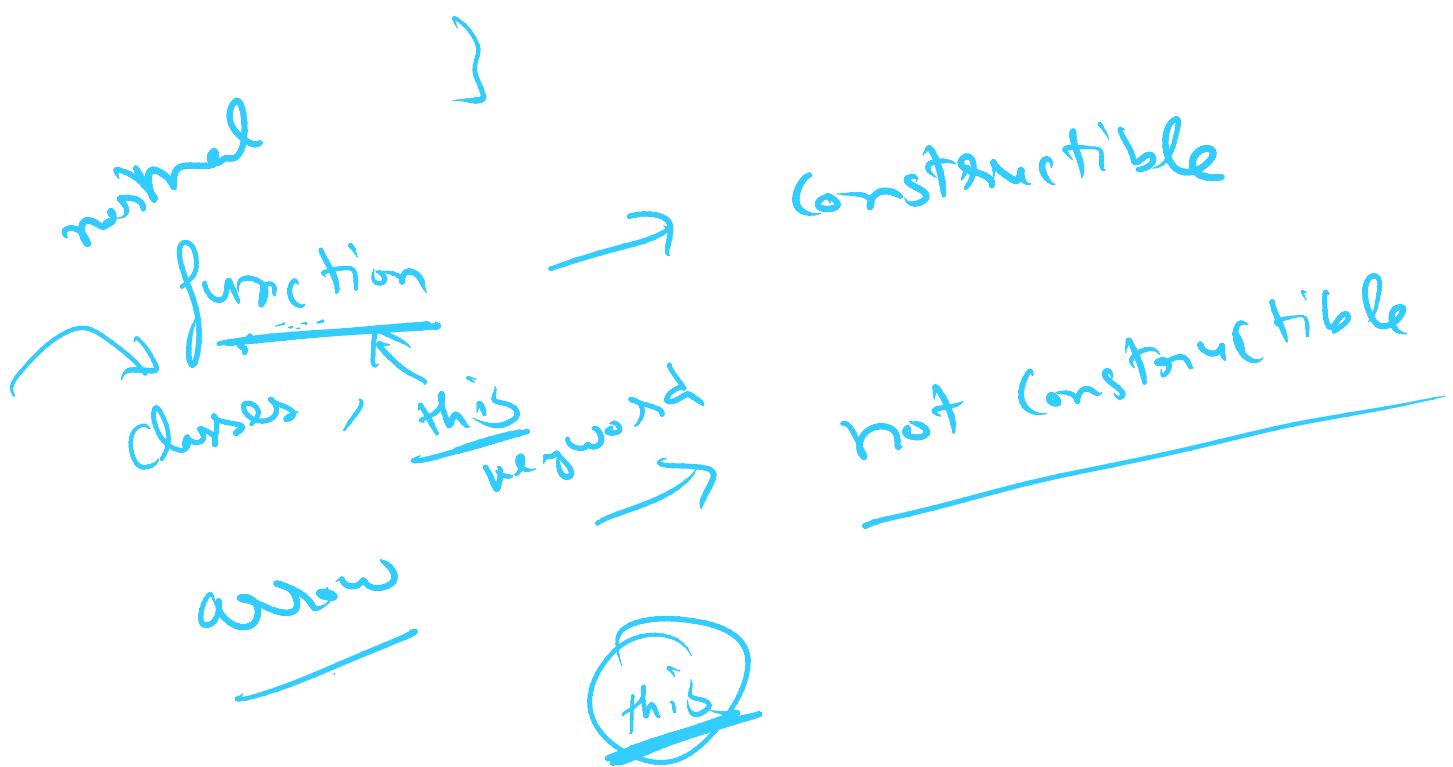
```

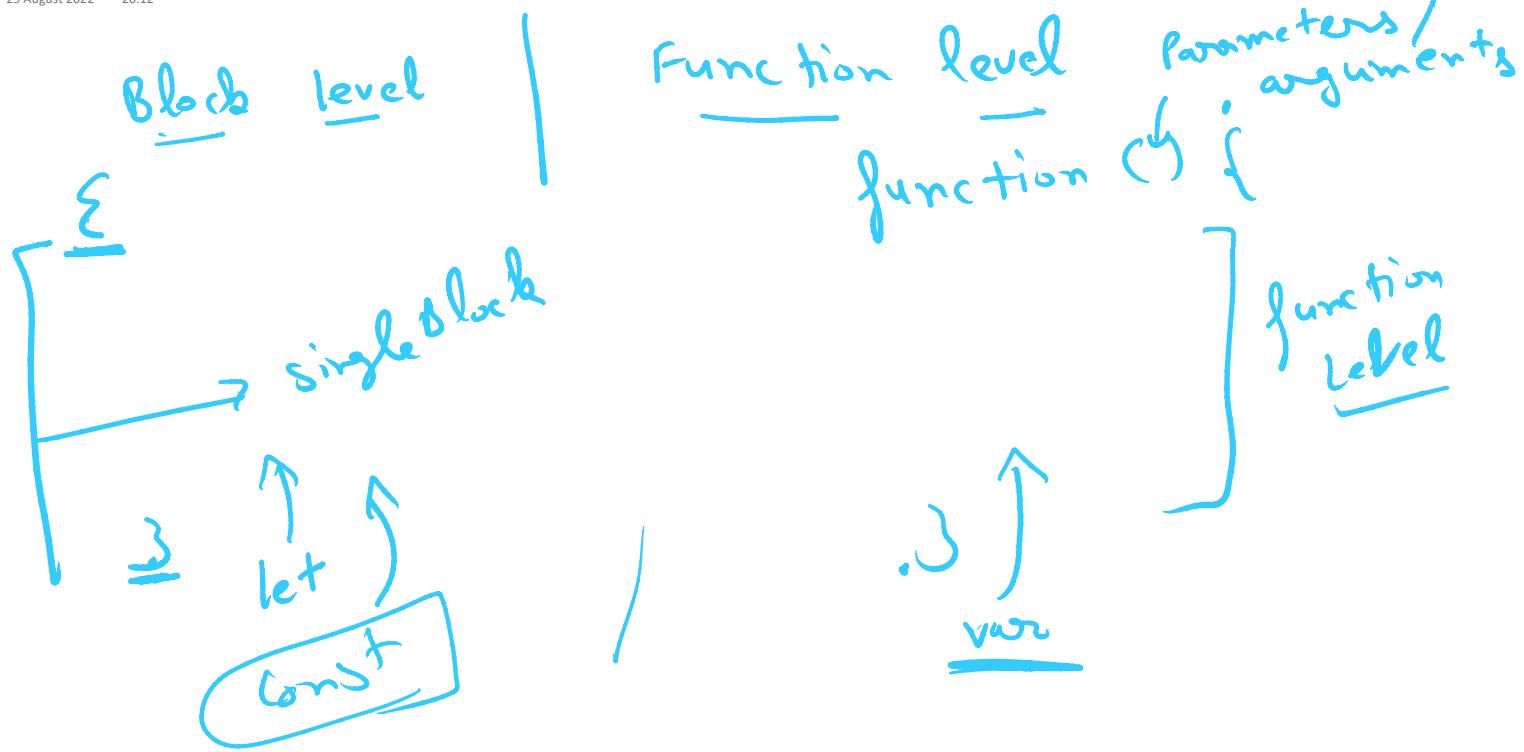
The functions which can be stored in a variable or can be passed as an argument to a function are known as first-order functions.



destructor

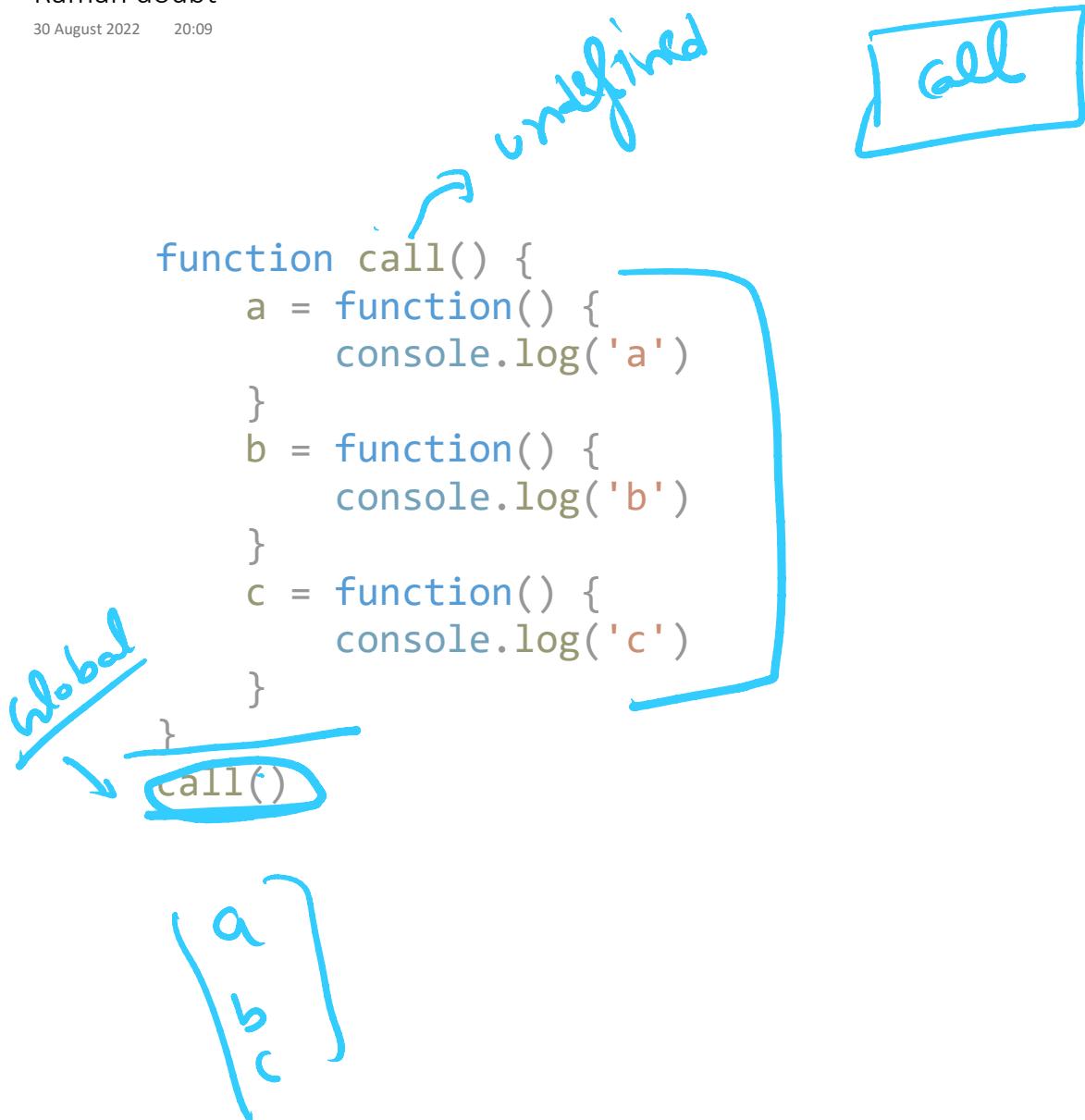
Const abc = ()^{parameters} $\Rightarrow \{$



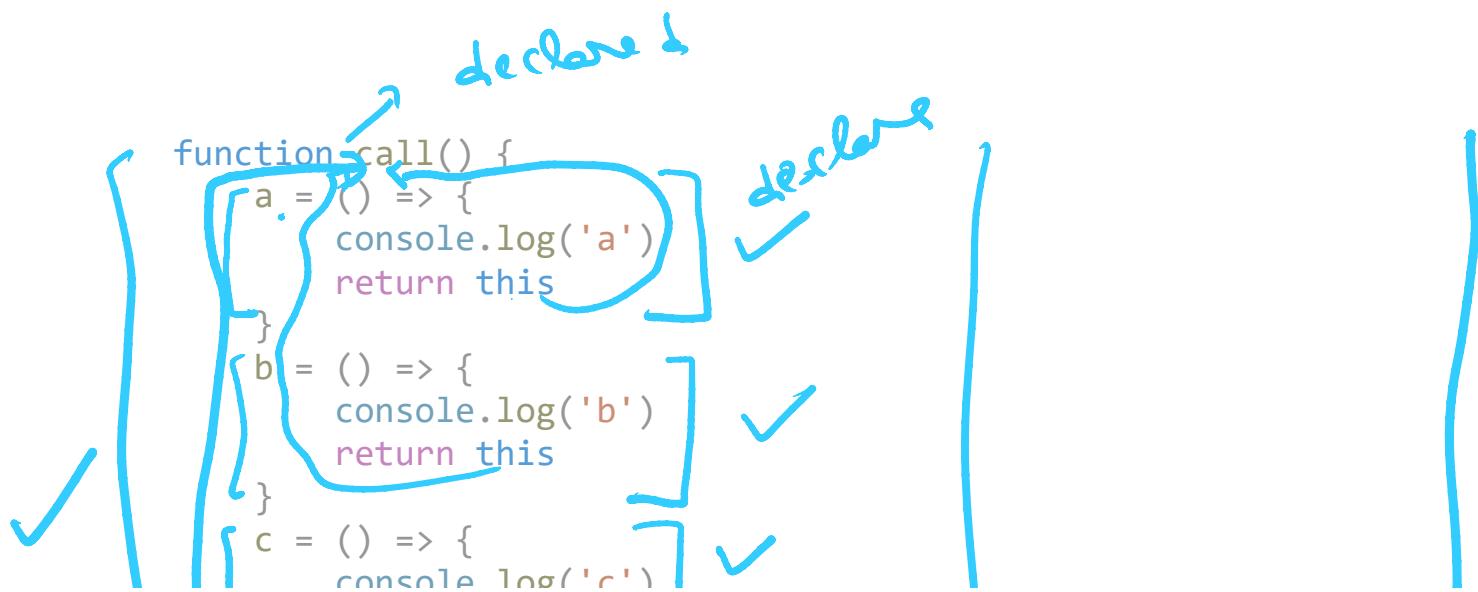


Raman doubt

30 August 2022 20:09



declare ↗

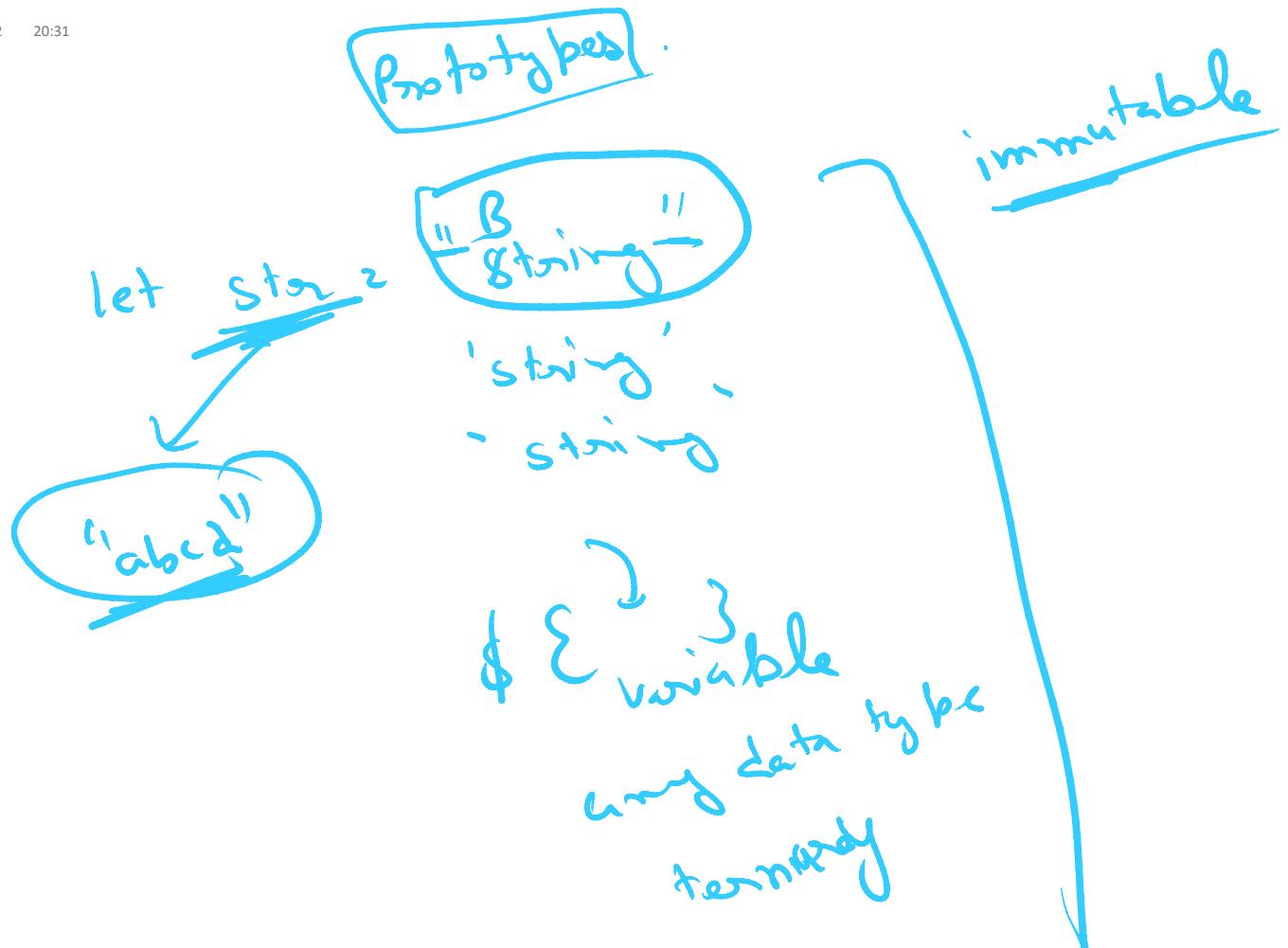


✓ | [{
} c = () => {
} console.log('c') }] ✓
} return this
}
call().a().b().c()
call
call
call
call
a
b
c
undefined

c()
b()
a()
call()

Strings

30 August 2022 20:31



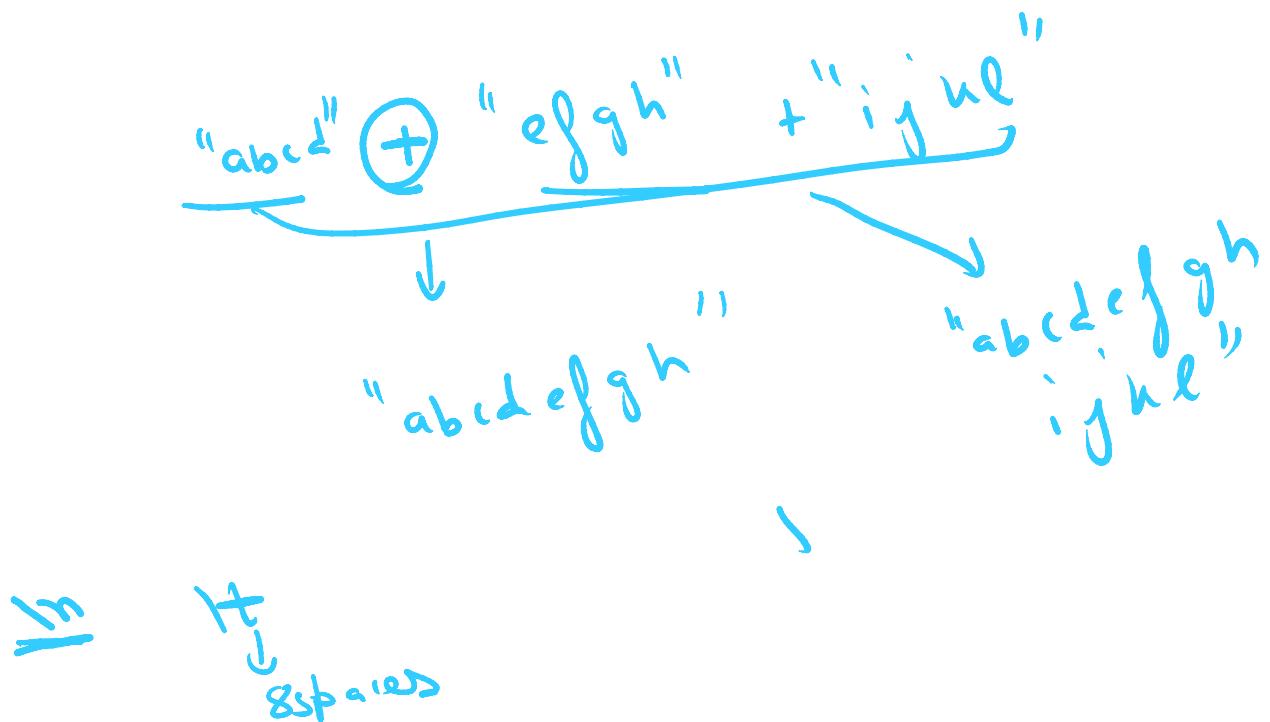
+ number
of arg.

function (---){

return {sum of
all the
arg.}

}

Concatenation of Strings



Methods on Strings

① length

let str = "string"

console.log(str.length)

↓
6

② str.charAt(_{index})
↓
a

String \rightarrow char

③ to Uppercase

str. to Uppercase ()

\downarrow
STRING

Substring

let str = "a b c d e f g h"
(S.I., No. of characters)

substr (S.I., E.I.)
substring (S.I., E.I.)
↓ don't get count +
last index

Str. substr(2, 4) \rightarrow cdef
str. substring(2, 6) \rightarrow cdef

Star. subsidiary 111 -

三

Substitution ($\frac{1}{S.I.}$, $\underline{\textcircled{3}}$)

let str = "string"
str
SubString (1, 4)
S.I E.I

b) Split
↓
String → Array

```
let star = "1,2,3,4,5"
```

Str. split ("_") → [1, 2, 3, ""]

str. split $\backslash - \backslash$

let str = "hey|h|ow|are|you"

str.split(" ") \rightarrow ["hey", "how", "are", "you"]

"hey how are you"



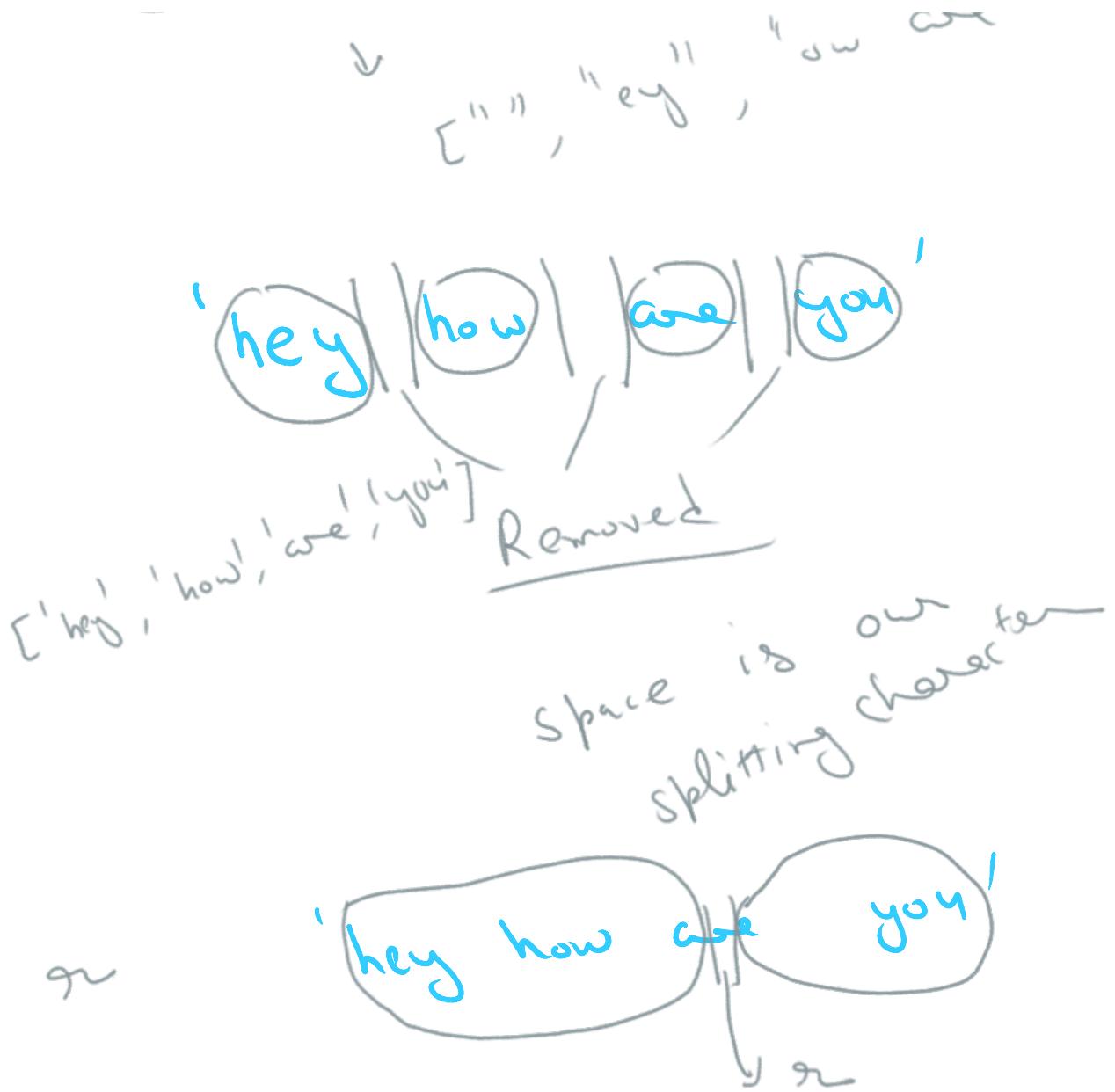
["", "ey", "ow are you"]

no. of lines + 1

• split("h")

"hey ow are you"

↓ "ou" "ow are you"]



\mapsto^2

['hey', 'how', 'are', 'you']

' ' ' '

8 trim

"hey how are you" " "

All spaces before the 1st character
All spaces at the end of the last character

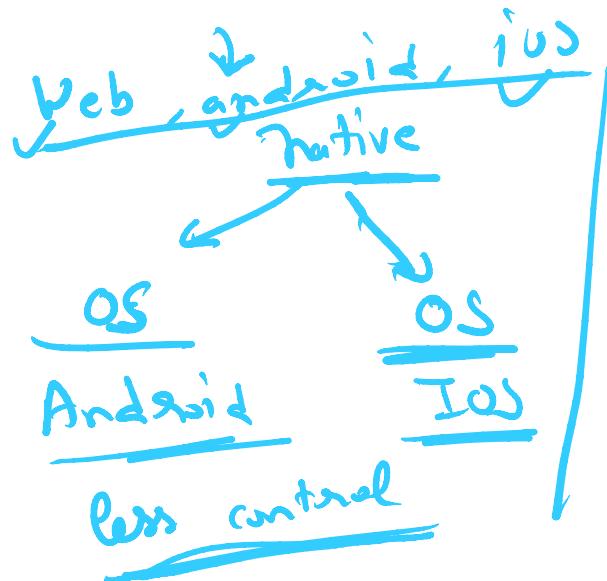
9 includes

let str = "hey how are you"

str.includes('how')
↳ boolean
true ↳ false

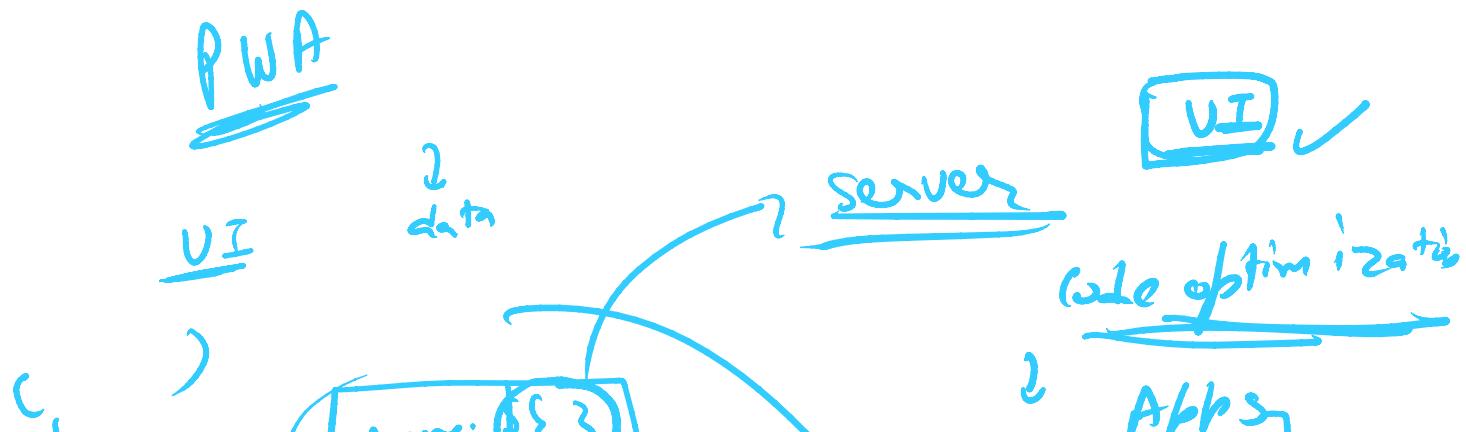
10. replace

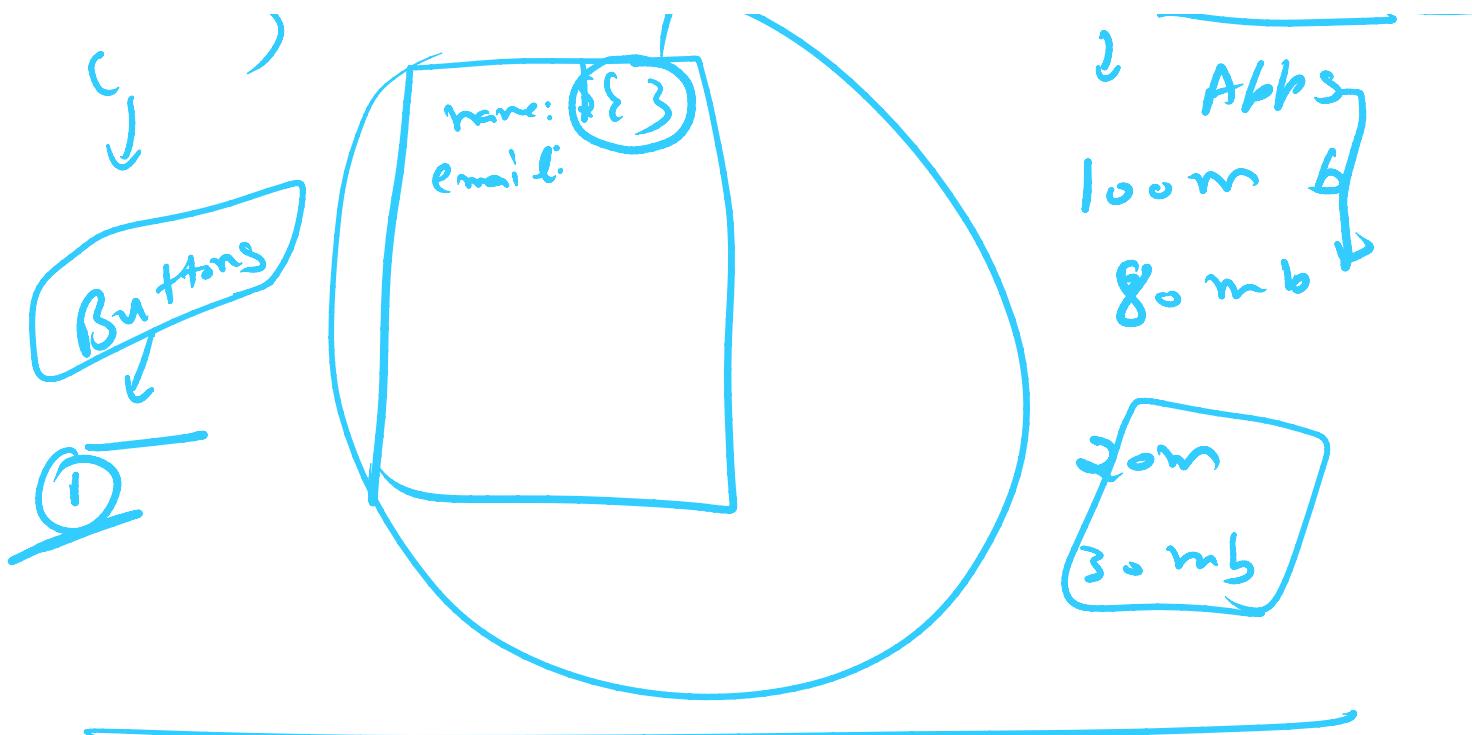
let str = " my name is variable"
str.replace("variable", "bharat")



Flutter

control
in built functions



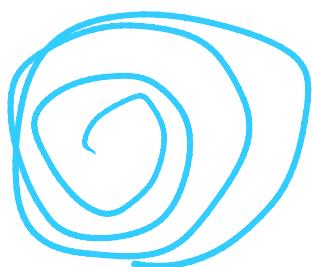
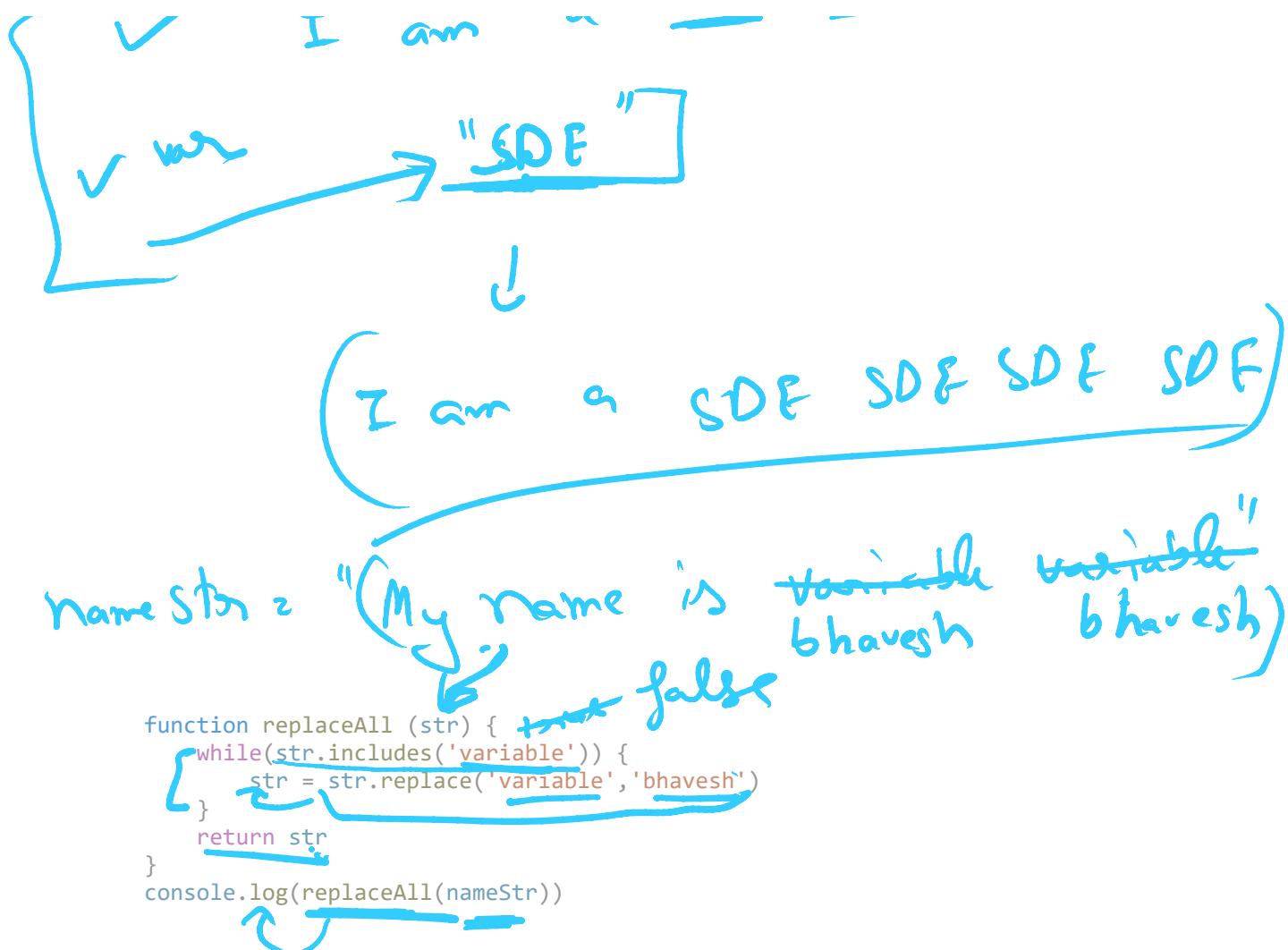


11. charCodeAt(0) \rightarrow index

character \rightarrow Ascii

let str = "abcd"
 str.charCodeAt(0)
 \downarrow
 97

"I am a Var Var. Var Var"



let str = "abcd"
 str.indexOf('c')
 ↓
 2

13

lastIndexOf

↓ 2

1 ← bcd

d
↓
3

let str = "0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 they how are you?
 16 17 18 19 20 21 22 23 24 25 26 27 28
 What about you?"

str.indexOf('you')

(12)

str.lastIndexOf('you')

(28)

14

concat

str1 = "abcd"

str2 = "efgh"

str3 = "ijkl"

str1.concat(str2) = "abcdefghijkl"

str1.concat(str2, str3)

Strl. concat(str1, str2)

↓
"abcd~~c~~ghijkl"

15 startsWith

str1 "My name is bhaveesh"

str.startsWith('my')

↓
false true

16 endsWith

str.endsWith('bhaveesh')

↓
true

x
**
@ * @
* * * -

$s \mapsto ' * '$

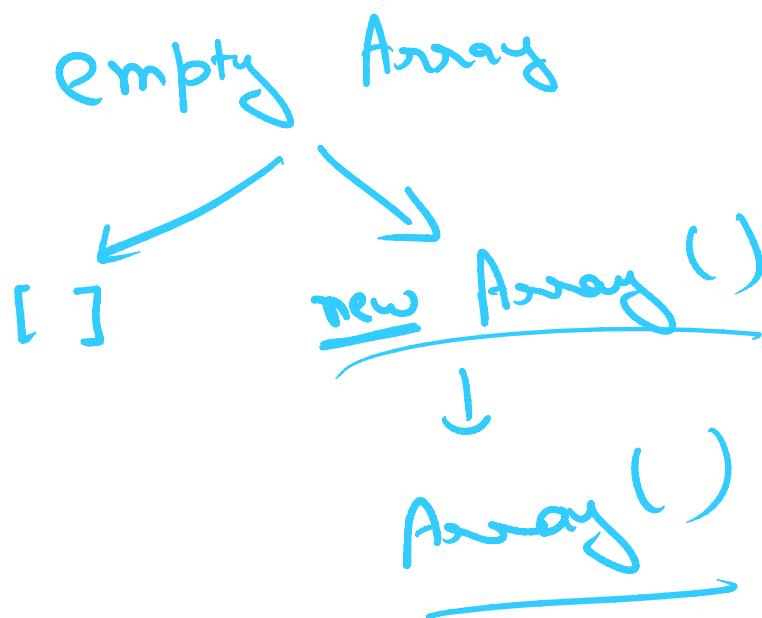
$s \mapsto s \cdot \text{repeat}(s)$

$' * * * '$

```
graph TD; s[s] -- "blue line" --> s_dot_repeat[s · repeat(s)]; s_dot_repeat -- "blue line" --> asterisks[' * * * ']; s -- "blue arrow" --> s_dot_repeat; s_dot_repeat -- "blue arrow" --> asterisks;
```

Arrays again

31 August 2022 21:33



Create an array with 8 empty
indexed

`new Array(8)`
`[-, -, -, -, -, -, -]`

Array Methods

01 September 2022 20:08

.length
↓

length of the array

[1, 2, 3, 4]

↓
4

last index
↓
3

(length)

let arr = new

Array(10)

lets say '*' →

(* * * *)

str.repeat(n)

[*, *, *] → n

CCCC

[PRACTICE]

Write a function to create an array filled with n number of '*'.

function (n) {

 return ['*' - ...n times] n

}

```
{ let arr = []
    function (n) {
        for (let i = 0; i < n; i++) {
            arr[i] = '*'
        }
    }
}
```

console.log(arr)

Let arr = Array(n).fill('*')

Concat

[1, 2, 3] { [4, 5, 6]}
.concat

↓
[1, 2, 3, 4, 5, 6]