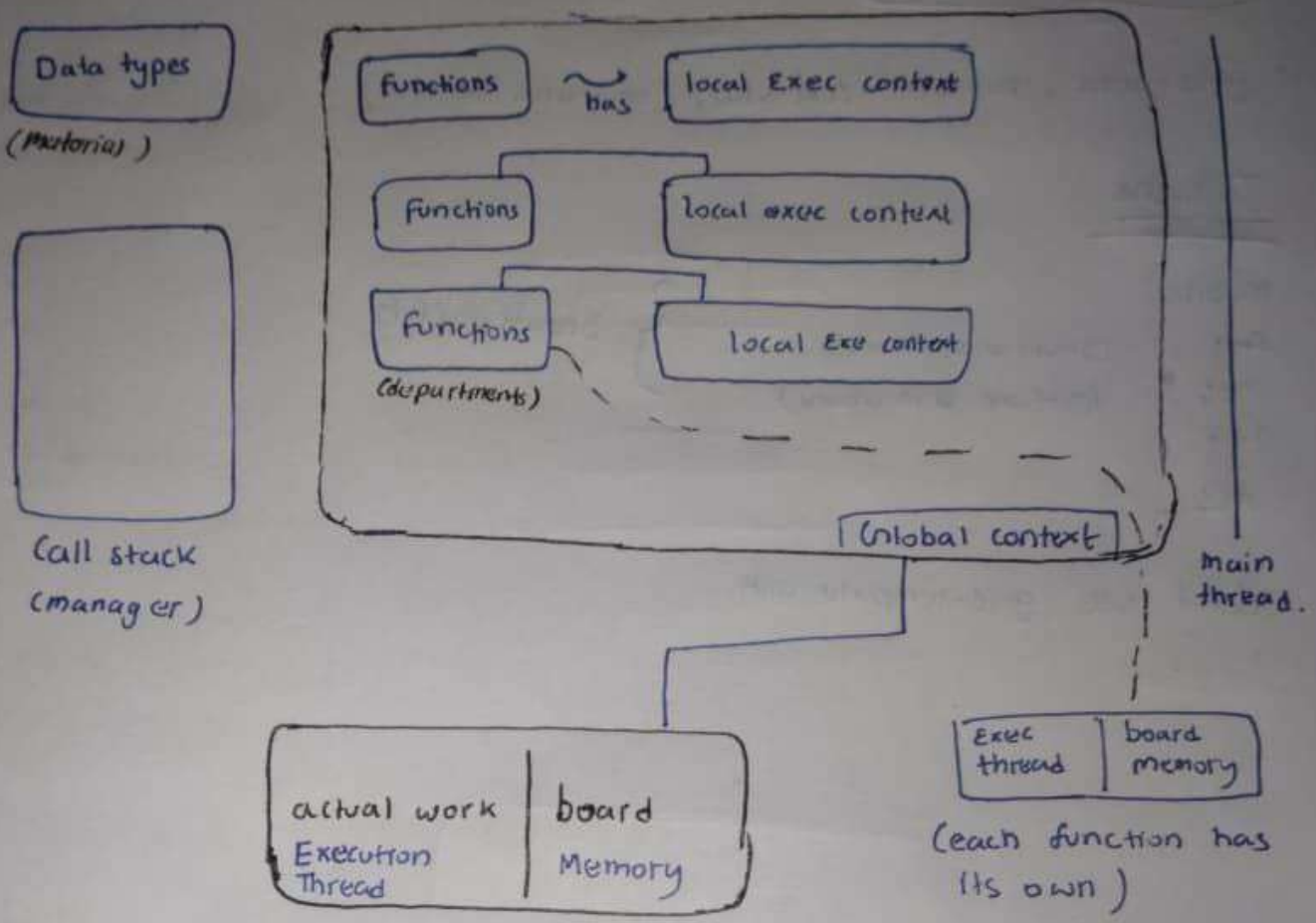
J-3

7 Feb | 26

JAVASCRIPT



(main thread : analogy → owner of factory, interpreter)

↓ reads line by line

> const number = 5 ⟶ stored in memory

> function addTwo (num) {

>

> } return num+2 → stored in memory

> const valueOne = addTwo (num) → function set to work (by call stack)

(main thread reads the function)

- nodeJS → runtime environment
  - ↳ not framework/Library

- node JS → V8 engine + Cpp compliances to move JS outta browser

- Internal JS → <script> // Your JS CODE </script>
- External JS → file_name.js
- JS is loosely typed language..

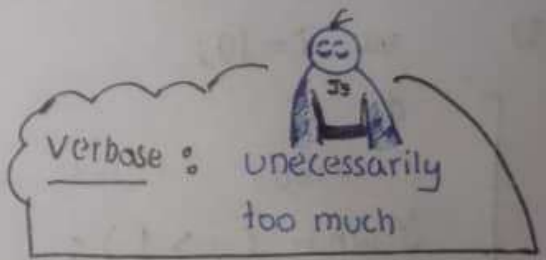

```
> function addNumbers (num1, num2) {
    var result = num1 + num2;
    console.log ('Result is', result);
  }

> addNumbers (3, 13) → go check console
```

It' okay to write
- var fname = "rgtu"
- fname = 36

CONDITIONAL CODE

→ returns boolean (yes or no)

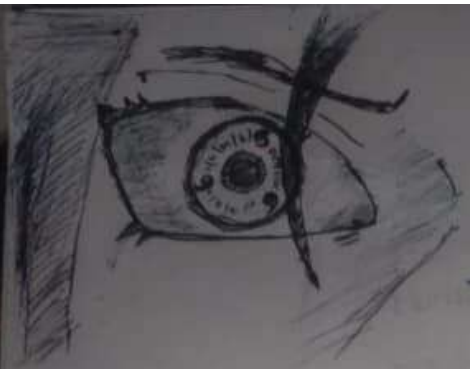
```
if (condition) {
  //execute this
}
else {
  // execute this
}
```



#multi-condition : var age

```
var childage = age <=12
var teen     = age <=19
var adult    = age <=40
var senior   = age > 40
```

```
if (childage) {
    // child
} else if (teen) {
    // teen
} else if (adult) {
    #adult
} else if (senior) {
    //senior
} else {
} //all false
```



Loop Is GENJUTSU

→ various Types of loops in JS

1) initialization | condition | updation

```
for (var x = 0 ; x<=10 ; x=x+1) {
    // code
}
```

→ when you exactly know no. of iterations you wanna have

---

2)

```
var filesize = 1024;
var currSize = 0;        → initialization.

while (currsize < filesize) {
    // Downloaded
}   currSize + 40;       // updation
```

→ When you know condition to stop, but not no. of iterations.

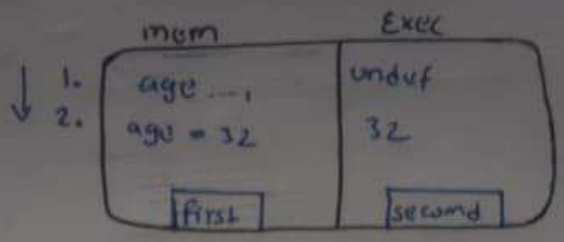
* Sirf kaam hone se matlab hai ..

---

3)

```
var i = 10;
do {
    // code , i = i-1   → updation
} while (i > 1);
```

→ ensures atleast one iteration




```
console.log (age)
var age = 32
console.log (age)
```

output ⇒ undefined
32

- age = 36
- console.log (age)
- var age

→ Hoisting : declarations (variables, functions, class) conceptually moved on the top of scope during compilation phase.

o/p : 36