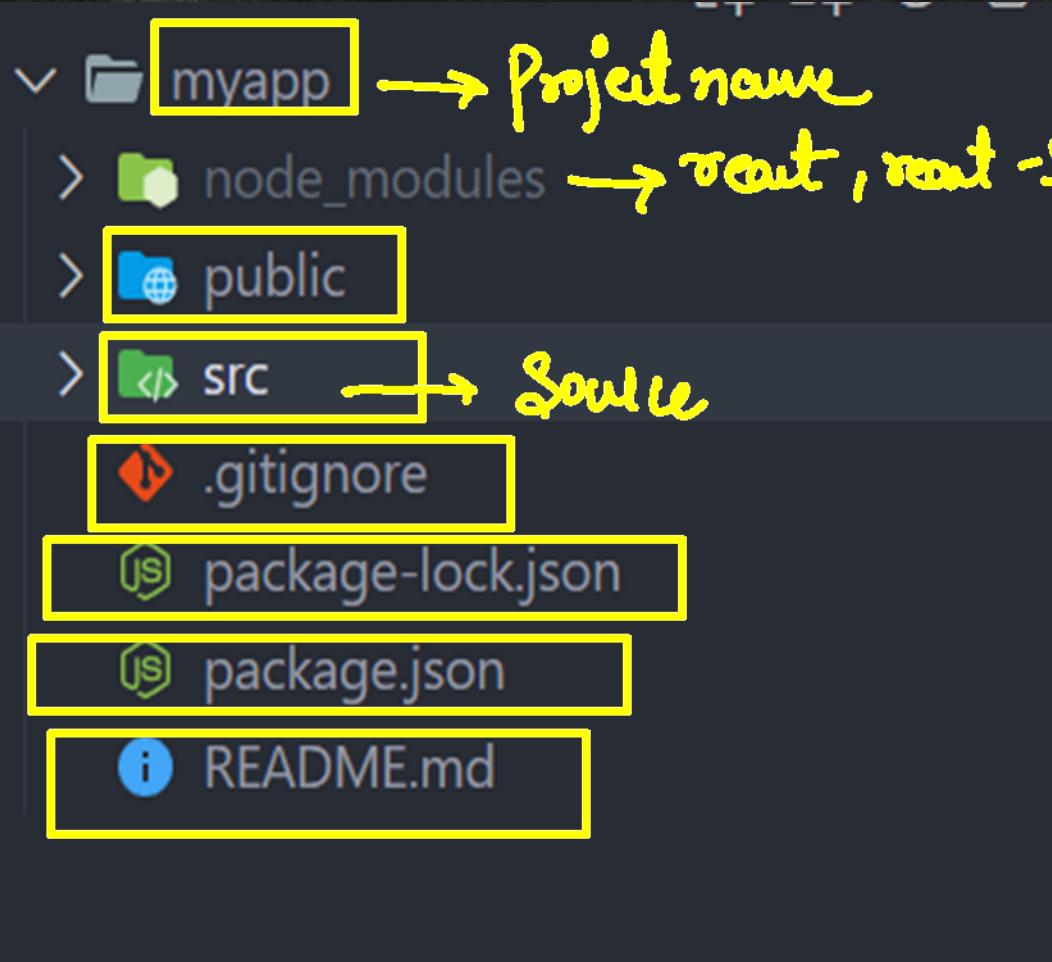


1) Node install

* 2) C/users/pc name/Appdata/Roaming/npm → Clash
 |
 | View
 | Show hidden
 | Content

3) Folder → npx create-react-app AppName → Docs
 |
 | Content
 | terminal
 |
 | Desktop

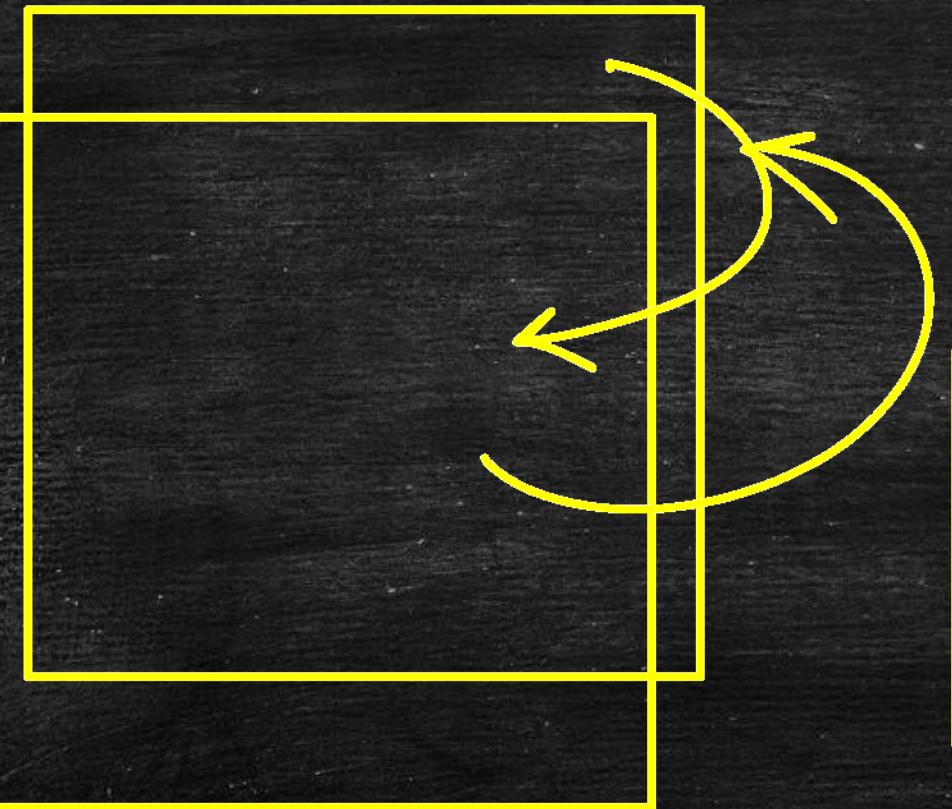
4) npm start



1) babel
2) auto

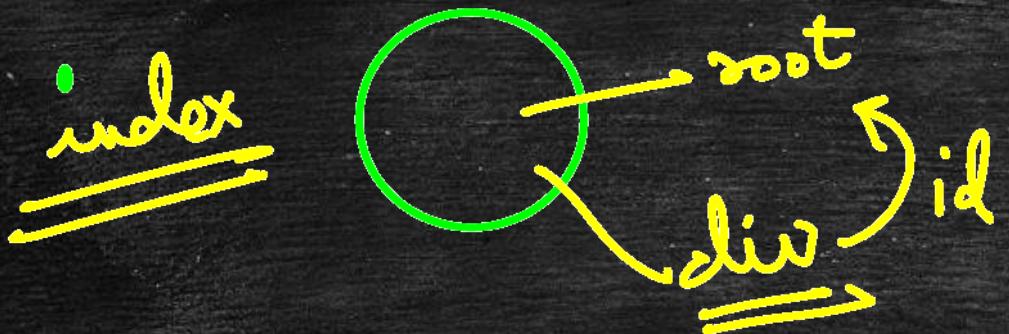
C# X

3) material-icons



$O(1)$

Node * = new Node();



let main
 = function main()
 { }
 ~~last main =~~
 { }
 Expression
 { }
main()

$() \Rightarrow \{ \}$

const app = $() \Rightarrow \{ \}$

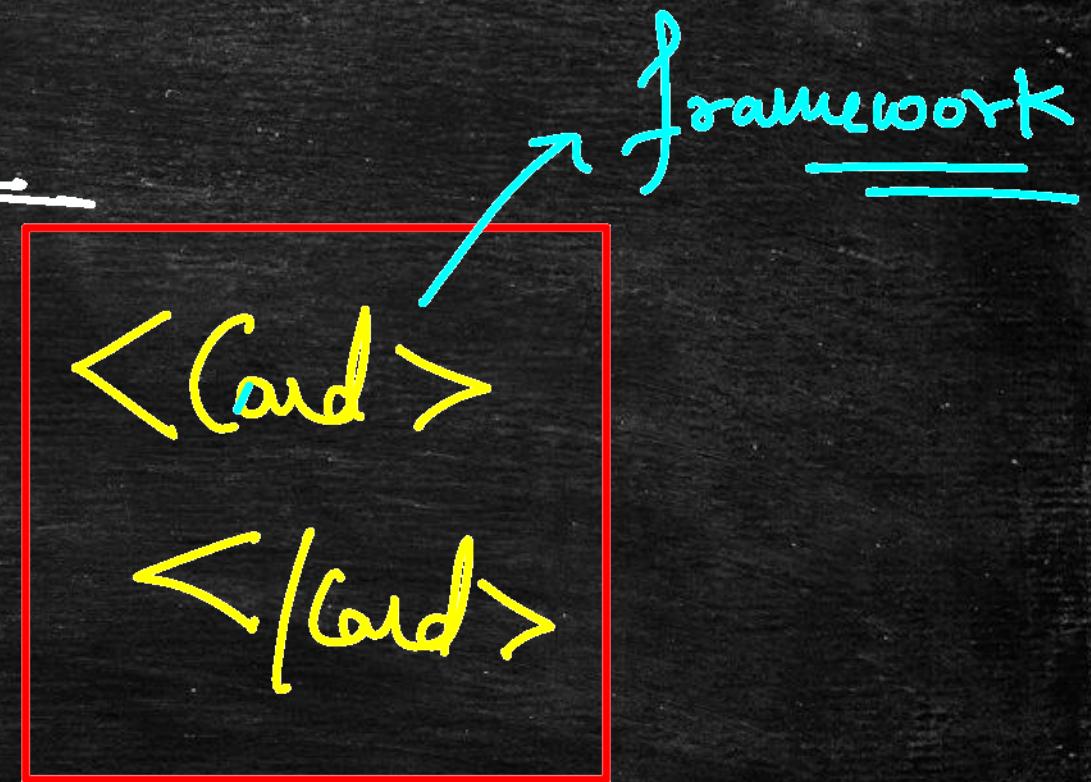
}

}

61 React 99
=====

↓
Components
=====

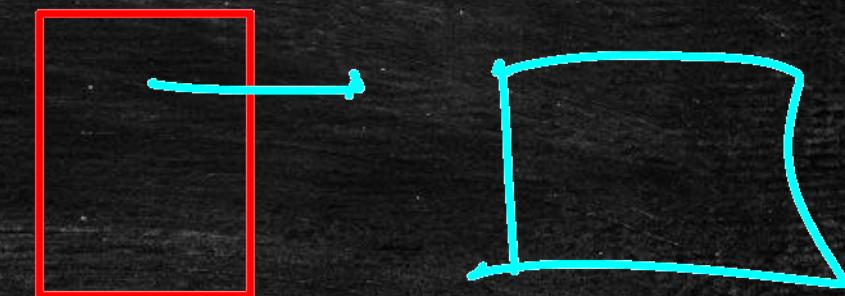
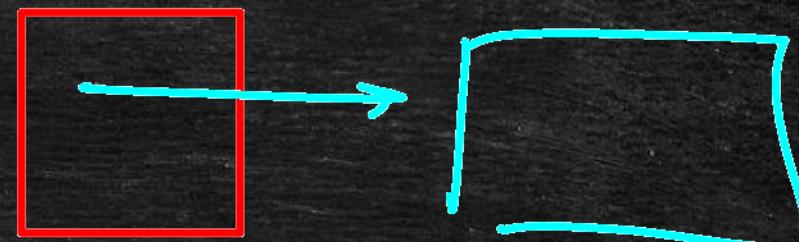
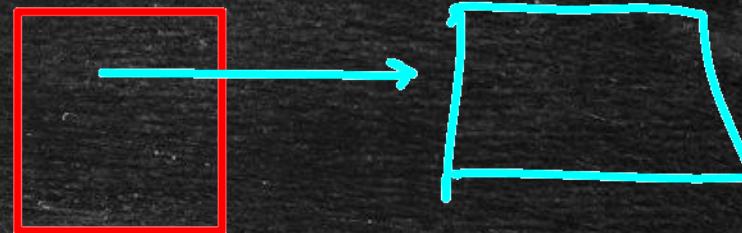
66 Not 19
Angular
=====



framework
=====

[$\{\text{f}3, \text{f}3, \text{f}3\}$]

array of objects



Const data = [{

},

{,

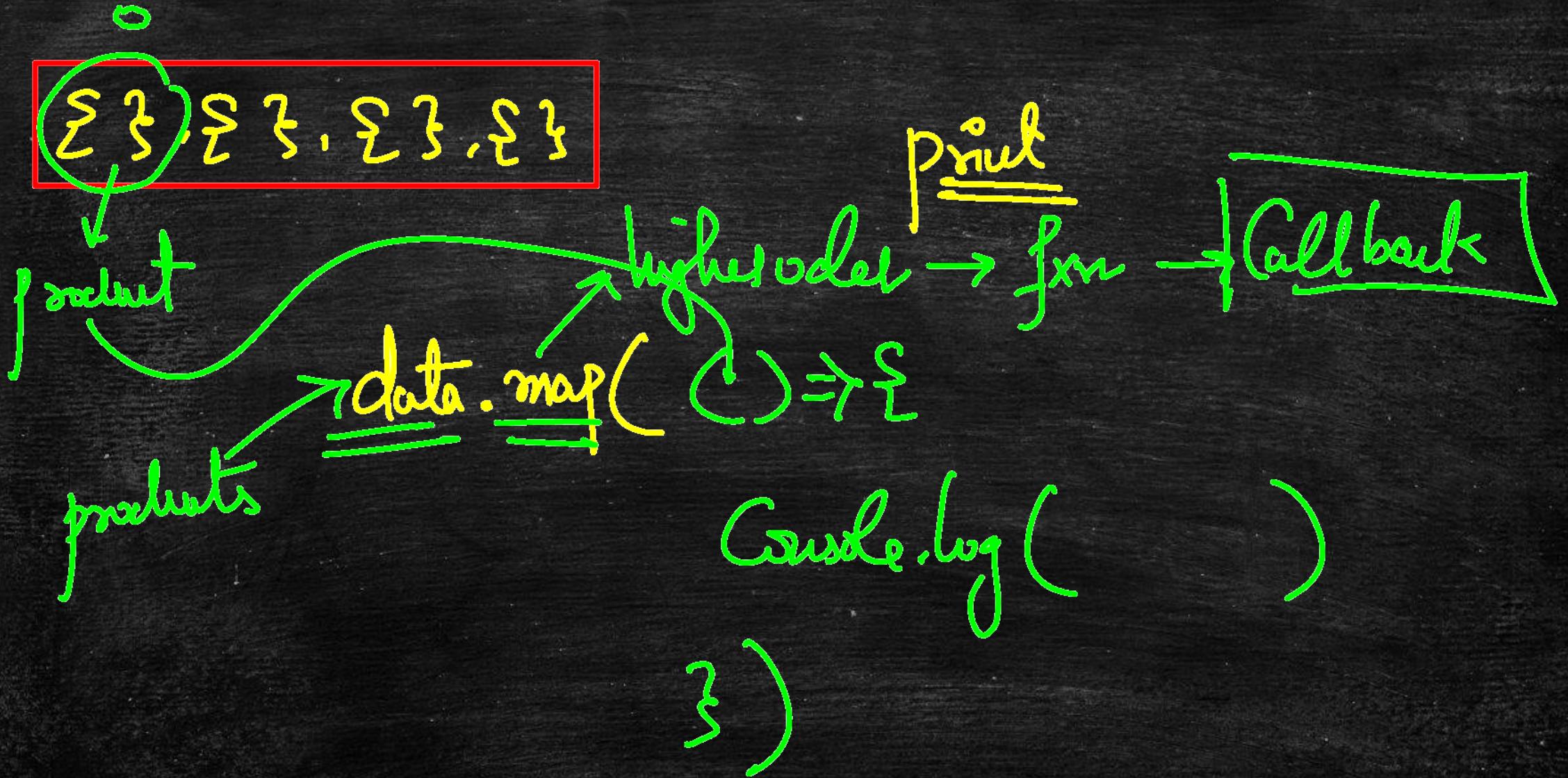
},

},

},

]

) data-structure



Event-driven / Asynchronous

Top - Bottom → Normal
Programs

high performance

Node

1) HTTP (Server)

2) limited
middleware

Express

1) Extended

HTTP

2) Comprehensive
middleware

3) Not a Web
framework

UI

3) Web application
framework.

APIs



- 1) Slope
- 2) Twisting
- 3) Re-assignment

99% -

Const

1%

let

Var

1) Scope

Block
Scope

Block
Scope

functional
Scope

2) Hoisting

Doesn't
allow

Does not
allow

Allow

3) Reassign → (cannot re-assign → de-assign & re-declared)

4) Global object	No	No	Yes
5) Temporal dead zone	Yes	Yes	No undefined

Console.log(a); — undefined
Var a;

functions

④ Named function :-

functions

Additi()

{

}

2) Functional - Expression

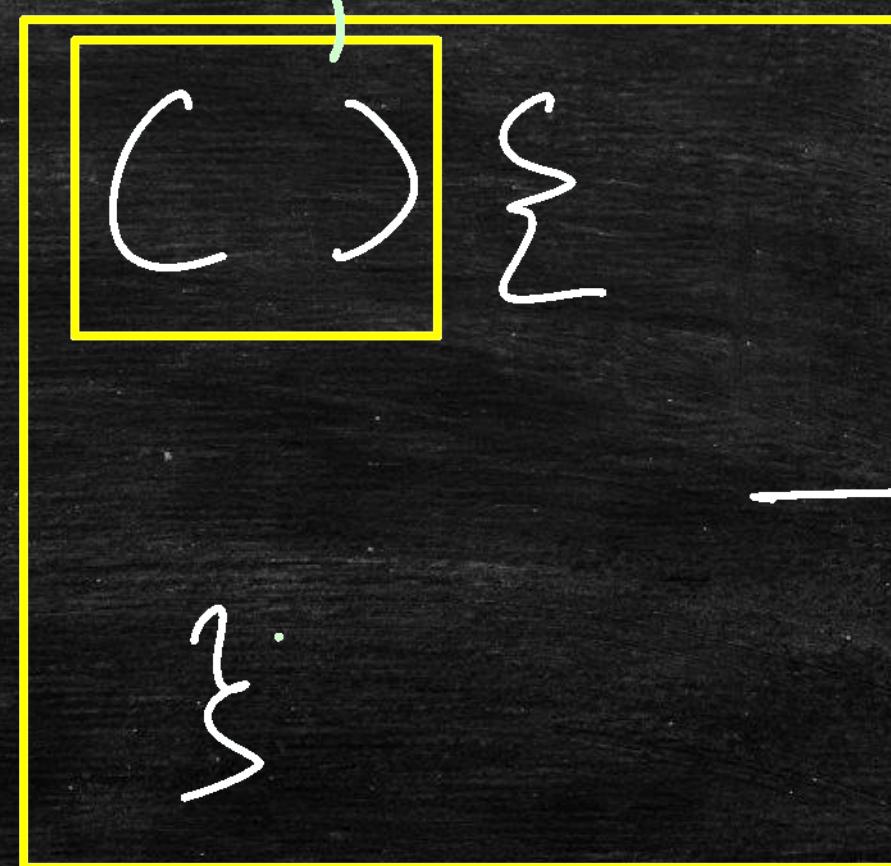
Const Add = Addit Σ

{

3) Anonymous Expression form

const Arri = [] {
};

Arri();



\hookleftarrow

$() \Rightarrow \Sigma$

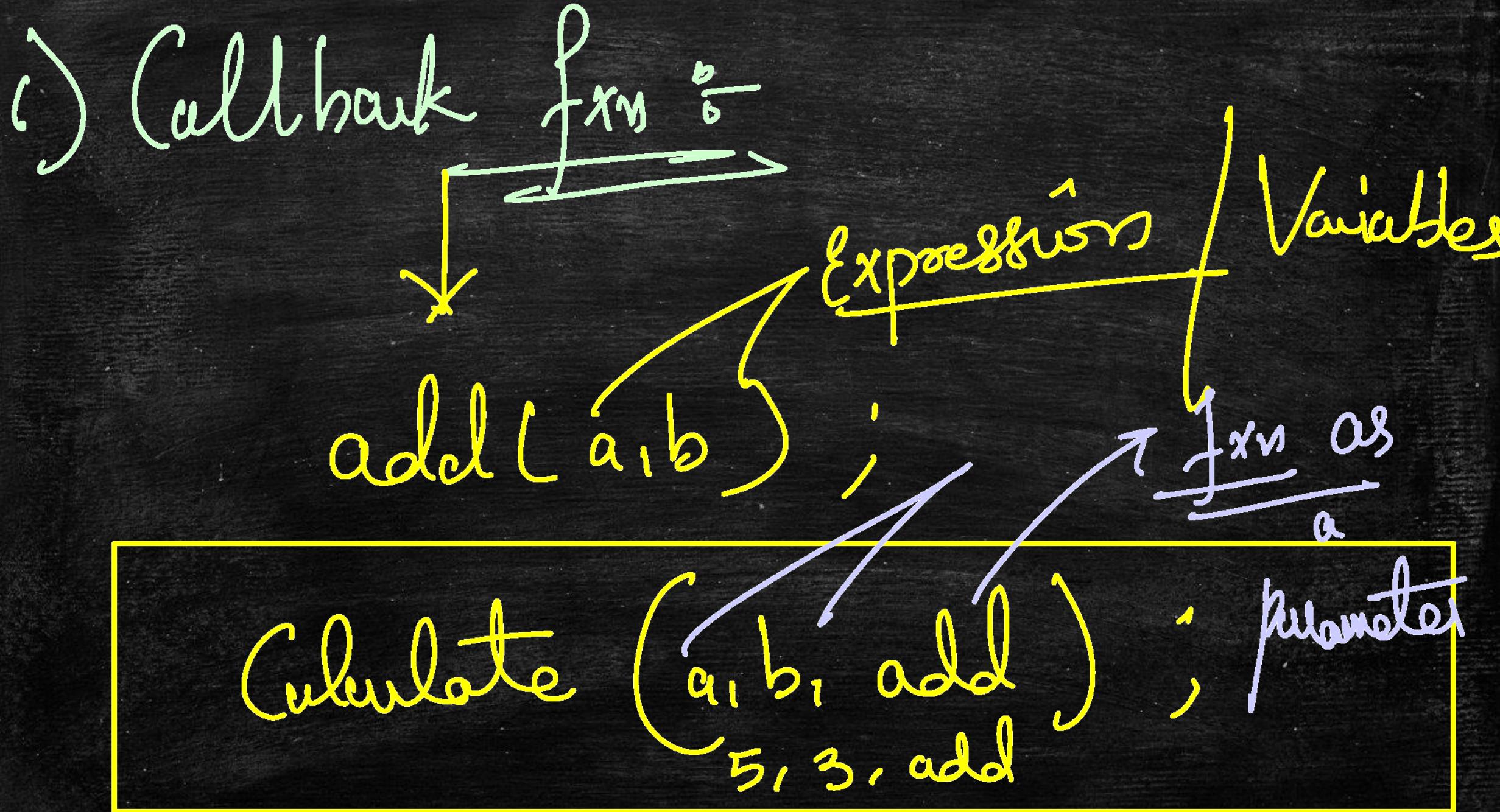
Arrow f^{x_m}

{ }

5) Expression Arrow \xrightarrow{fxn} :-

Const Adi = () $\Rightarrow \Sigma$

}



Cost₁ Calculate = $\langle a, b, \text{operation} \rangle \Rightarrow$

\sum
~~operation(a, b);~~
add

Cost₁ add = $\langle a, b \rangle \Rightarrow \sum$
 $\underbrace{\text{return } a + b;}_3$

~~7) Callback hell~~

~~process → event-driven~~

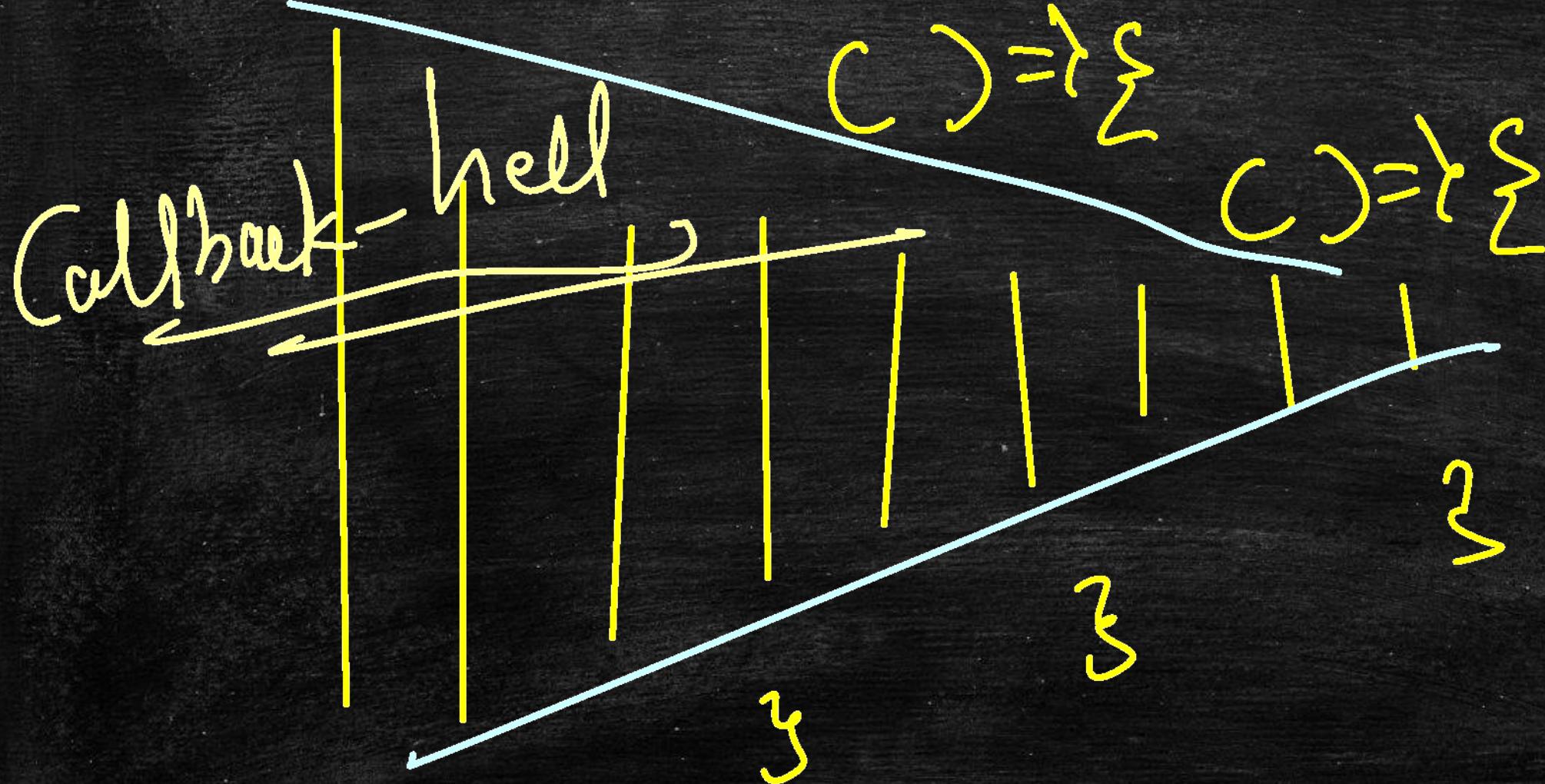
- 1) Regex
- 2) Enroll
- 3) Linux
- 4) Git
- 5) C
- 6) front
- 7) MERN
- 8) Certif

Const Regex = (a,b,()) =>

Σ () => Σ () => Σ

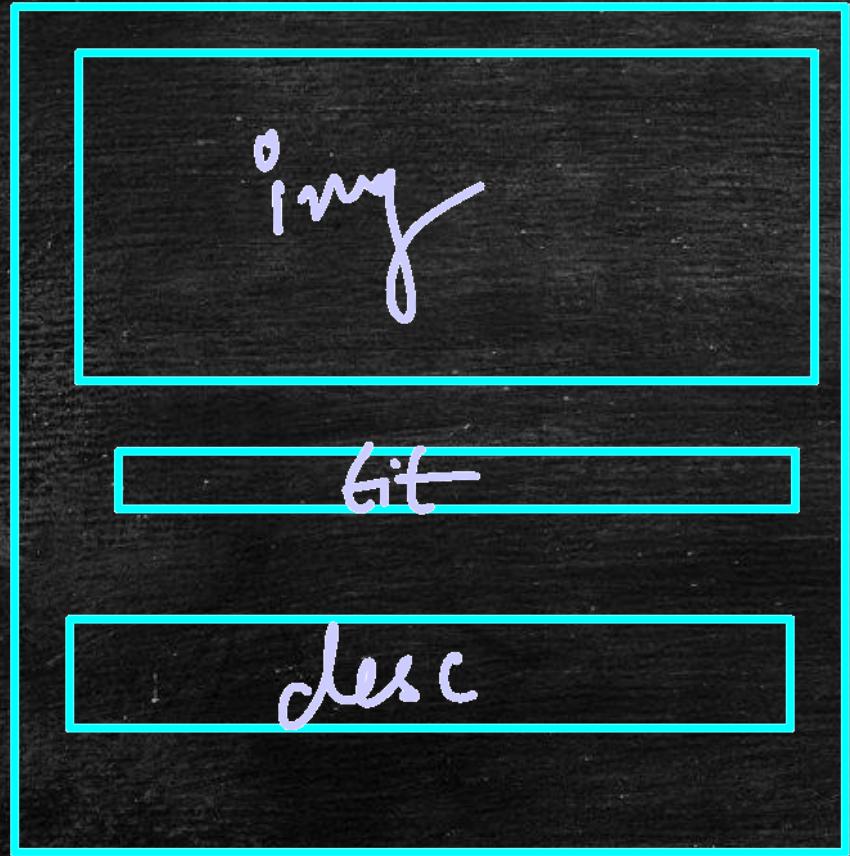
} ; }

$C \rightarrow \Sigma$



2) Promise

→ pending
→ resolve
→ reject

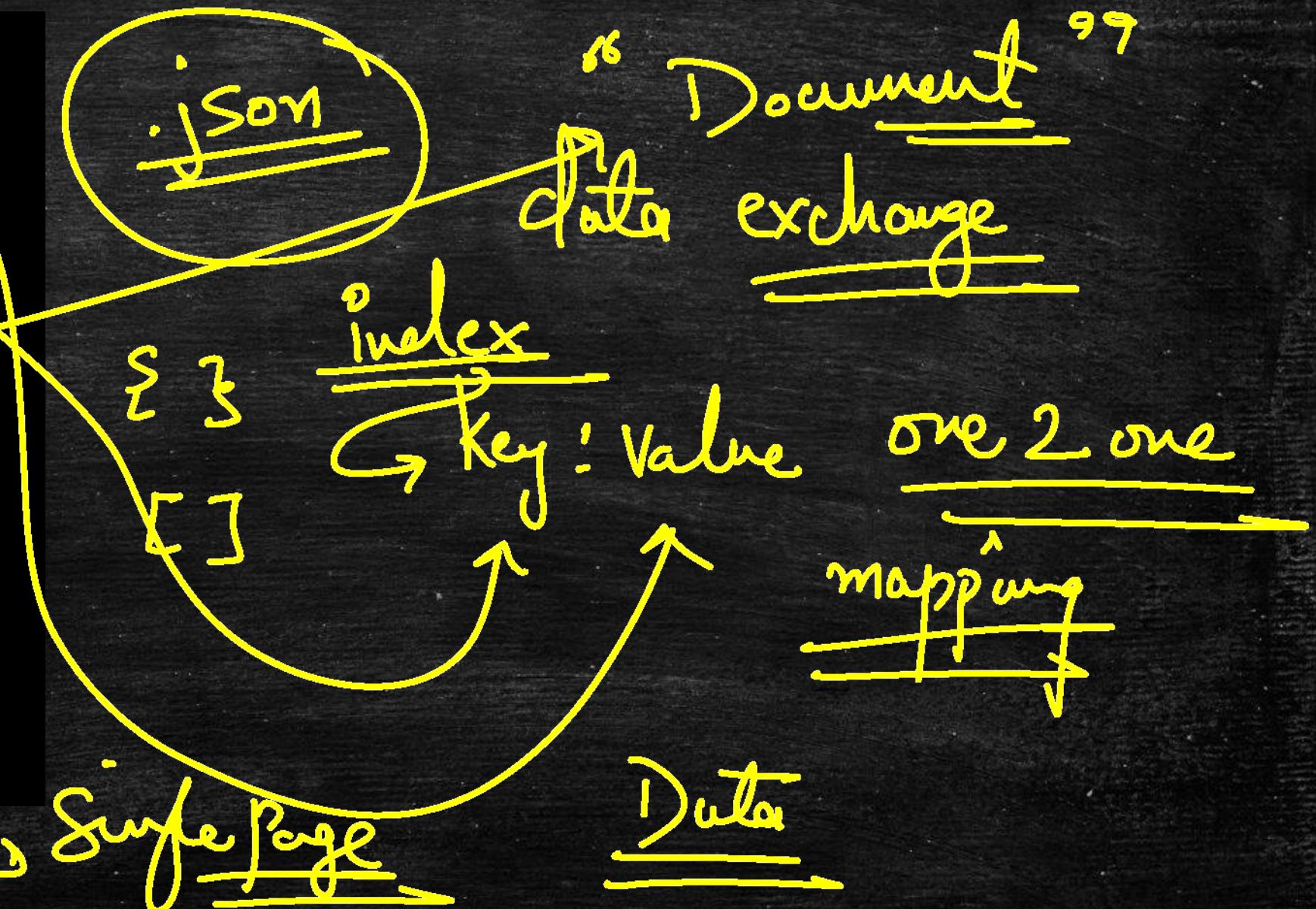


`empty Card`

`data-file`

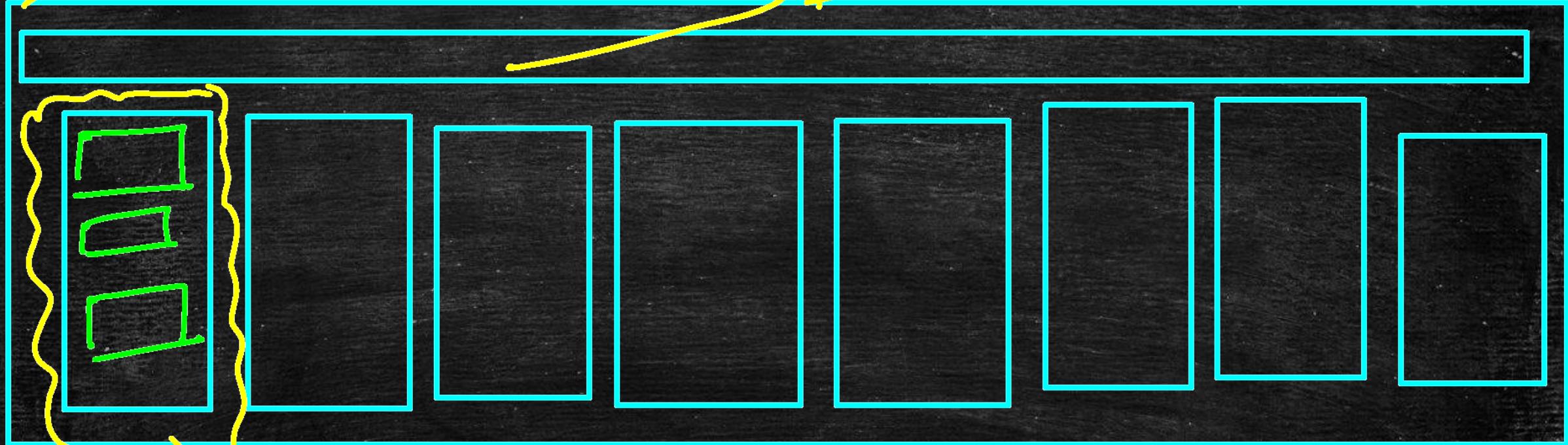
`card template`

```
{  
  "firstName": "John",  
  "lastName": "Doe",  
  "age": 30,  
  "isStudent": false,  
  "address": {  
    "street": "123 Main Street",  
    "city": "Anytown",  
    "zipcode": "12345"  
  },  
  "phoneNumbers": [  
    {  
      "type": "home",  
      "number": "555-1234"  
    },  
    {  
      "type": "work",  
      "number": "555-5678"  
    }  
  ]  
}
```



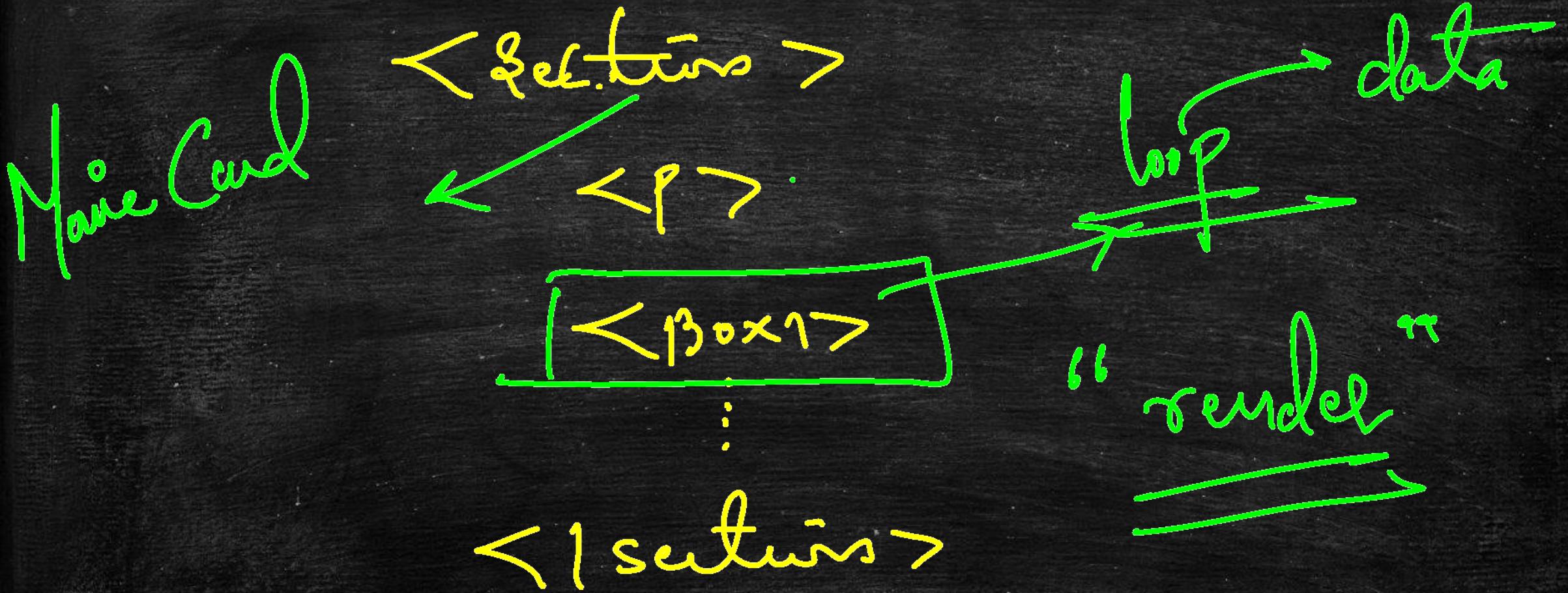
Parent → Sections

Item



Card

1 Card



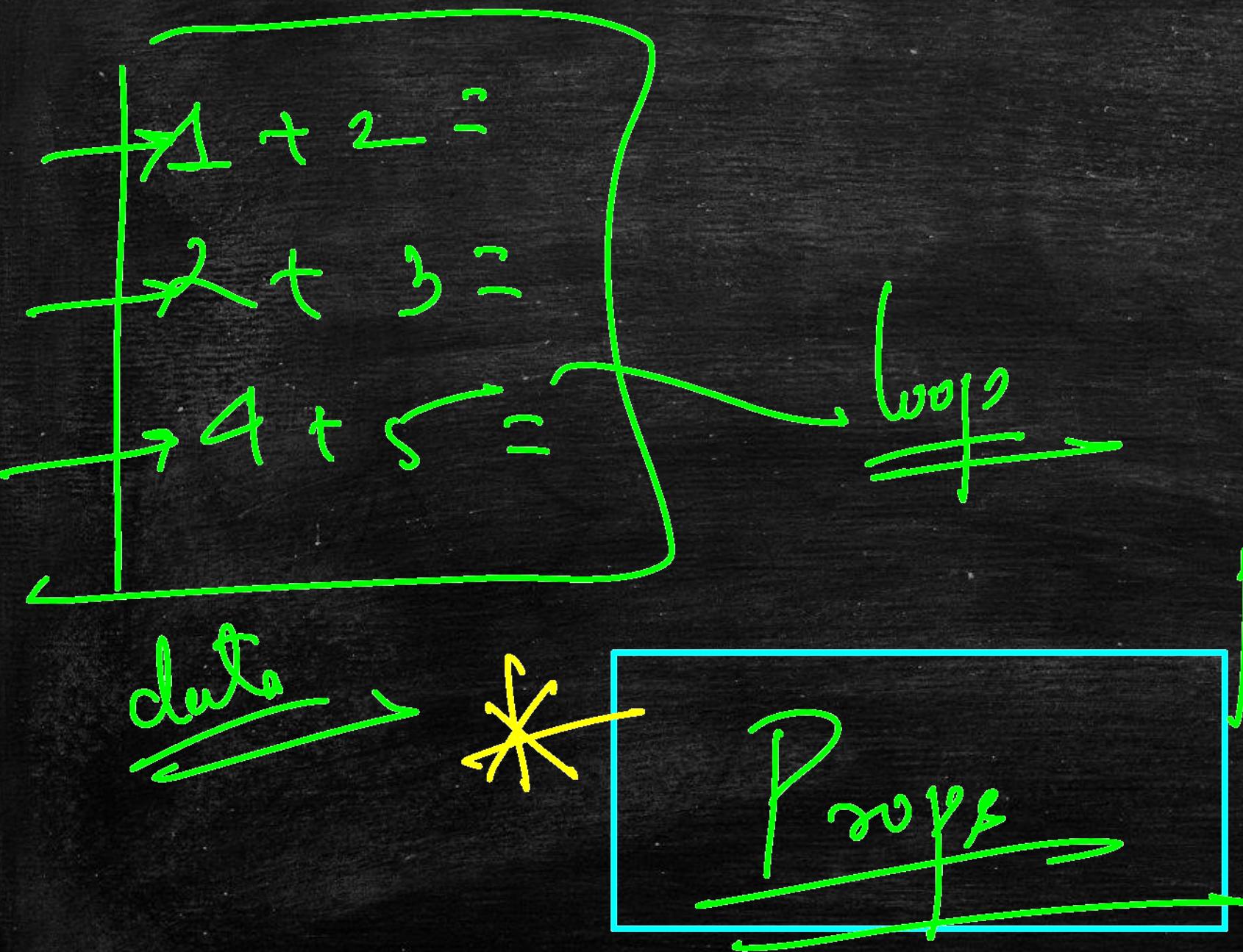
```
const data = [
```

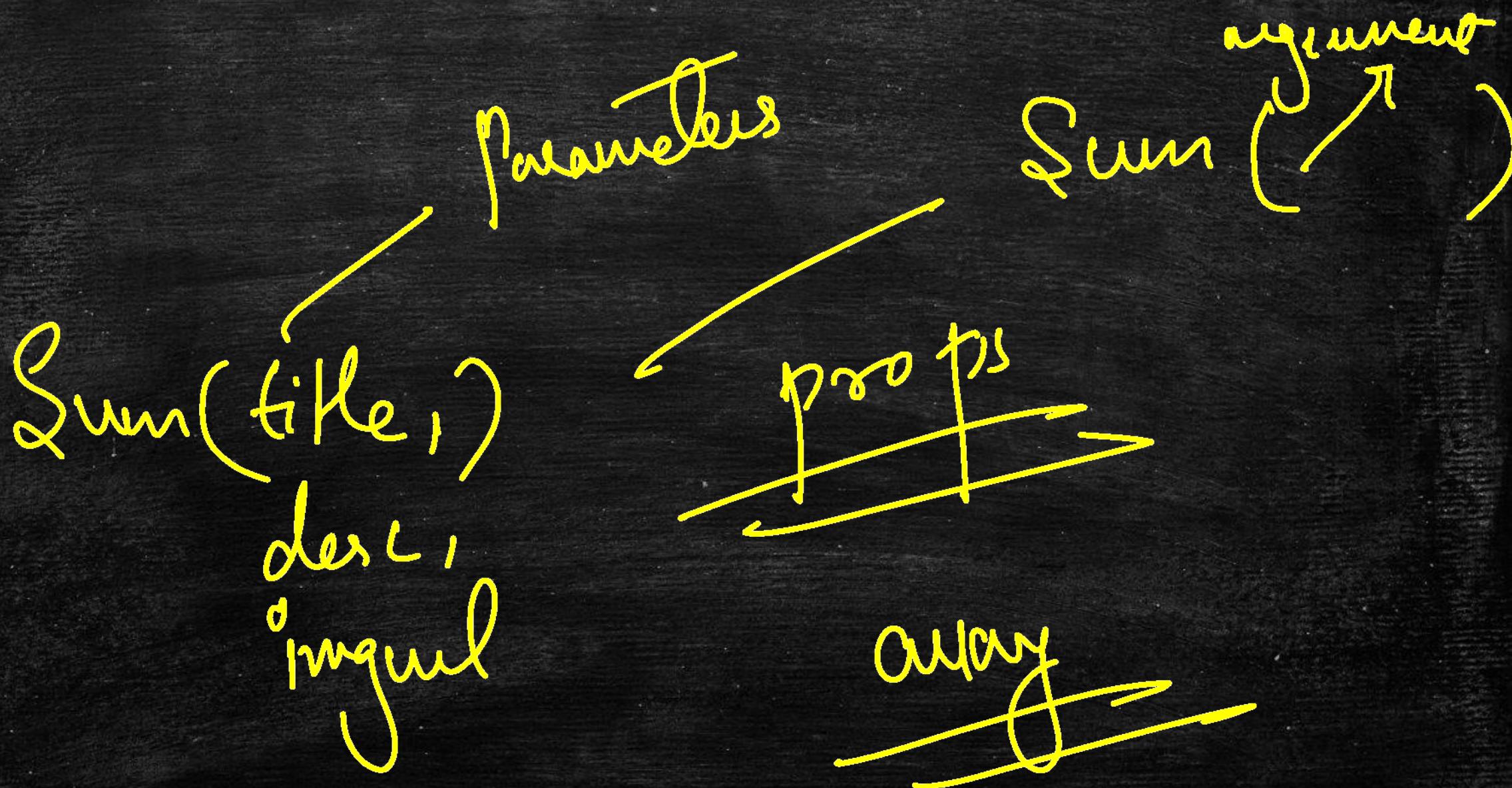
```
  { id: 1  
    title: _____  
    desc: _____  
    img: _____  
  },
```

parameters

Sum(a,b);

Parsing parameters





~~Props~~

const props = ['a', 'b'];

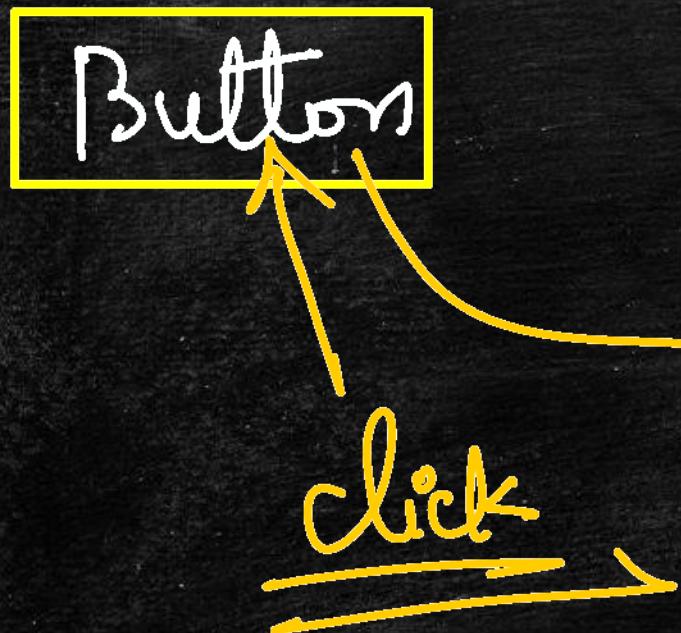
↓
~~destructuring~~

$$\{ 'a', 'b', 'c' \} =$$

de-structuring

2) Hooks:

env
↓



Variable → changes ko dekhi
nahi kai pata
hai.

⁶⁶
State → Paristhi → State Variable

= useState();

initial-value;

