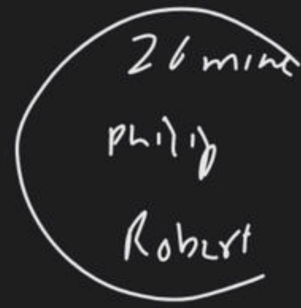
DOM + Modern JS - Class IV

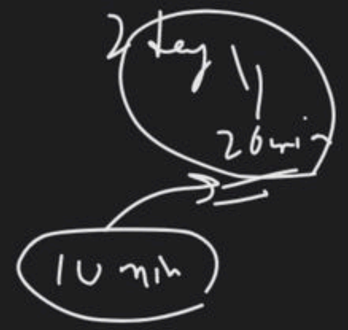
Special class

Love Babbar • Feb 21, 2023

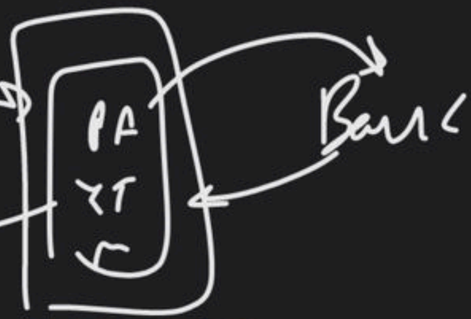
→ Asynchronous JS:-

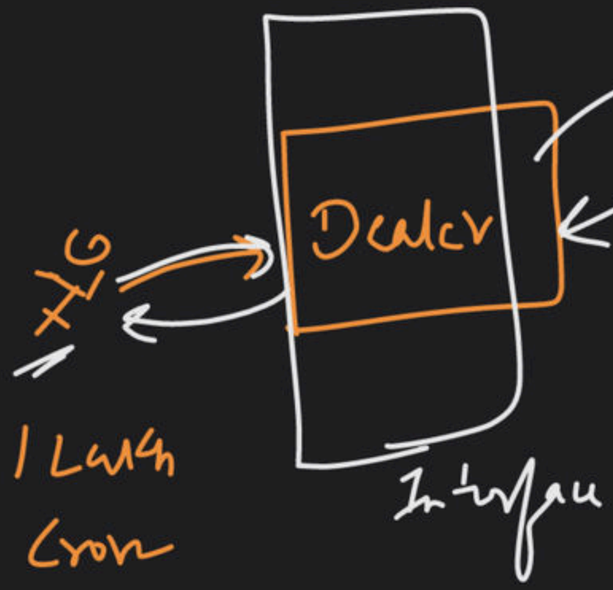
JS → Single threaded

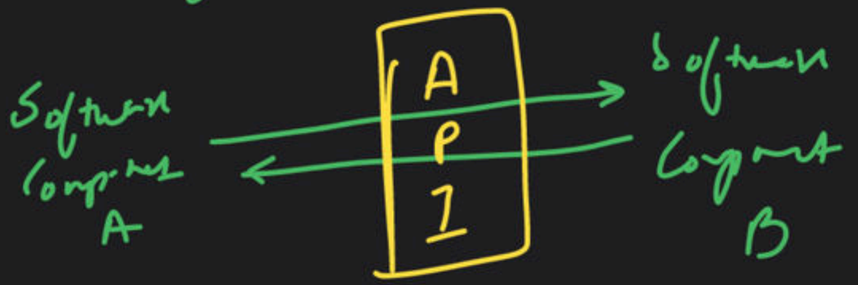


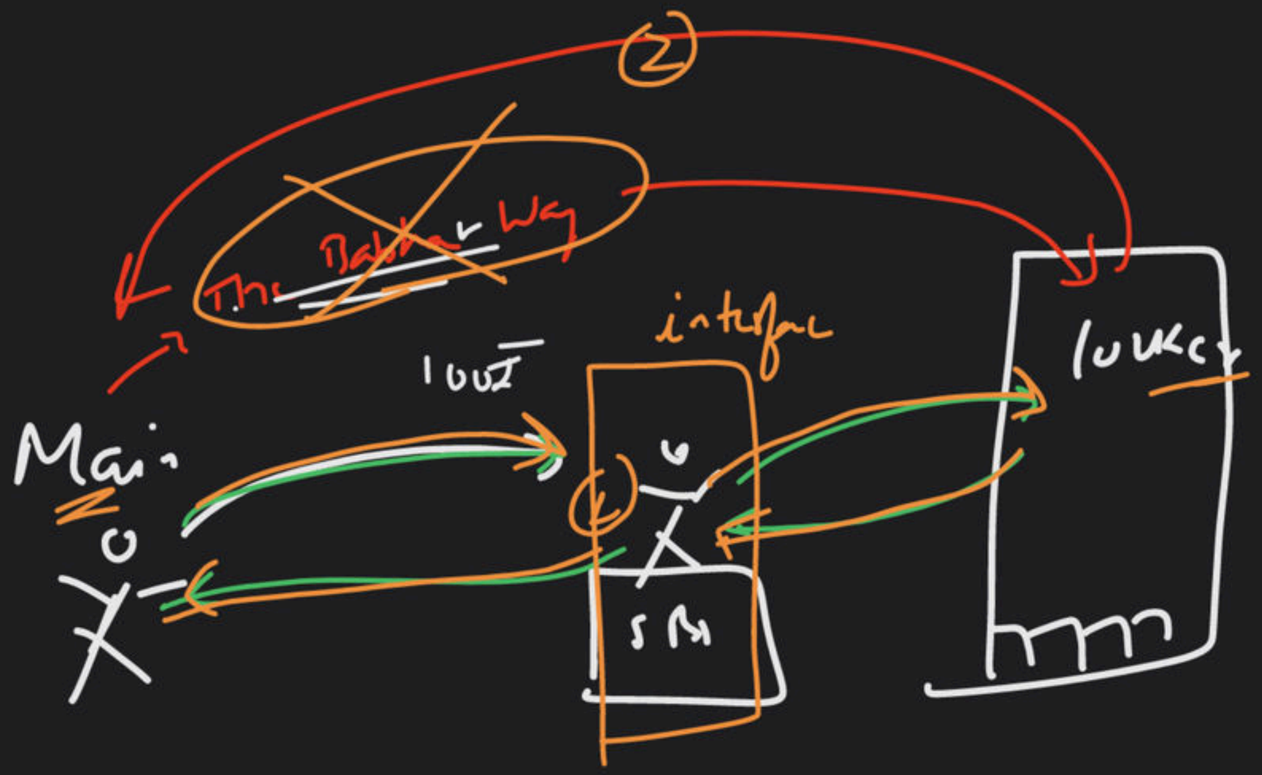


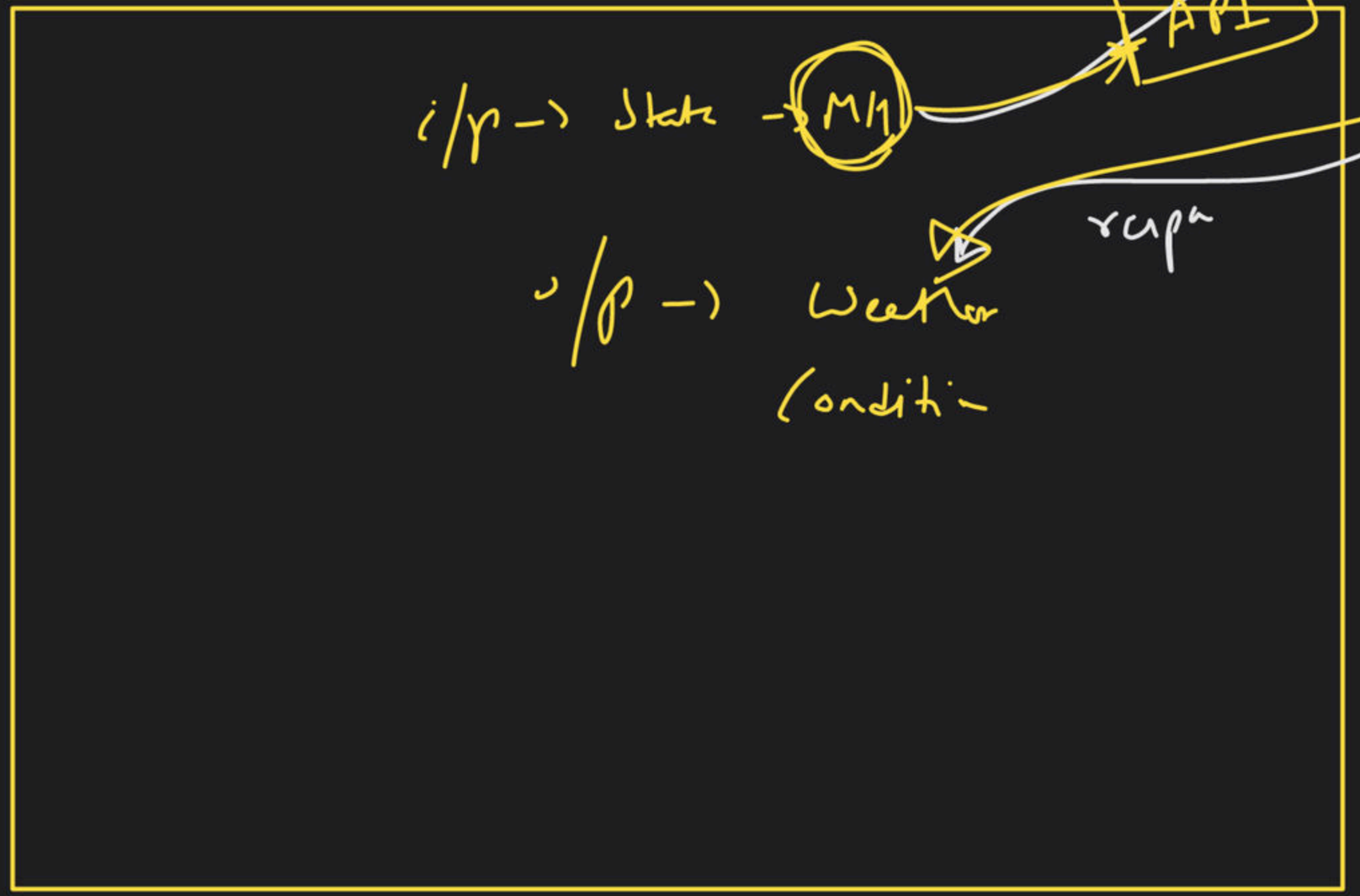
→ API :- application programming Interface











Weather App

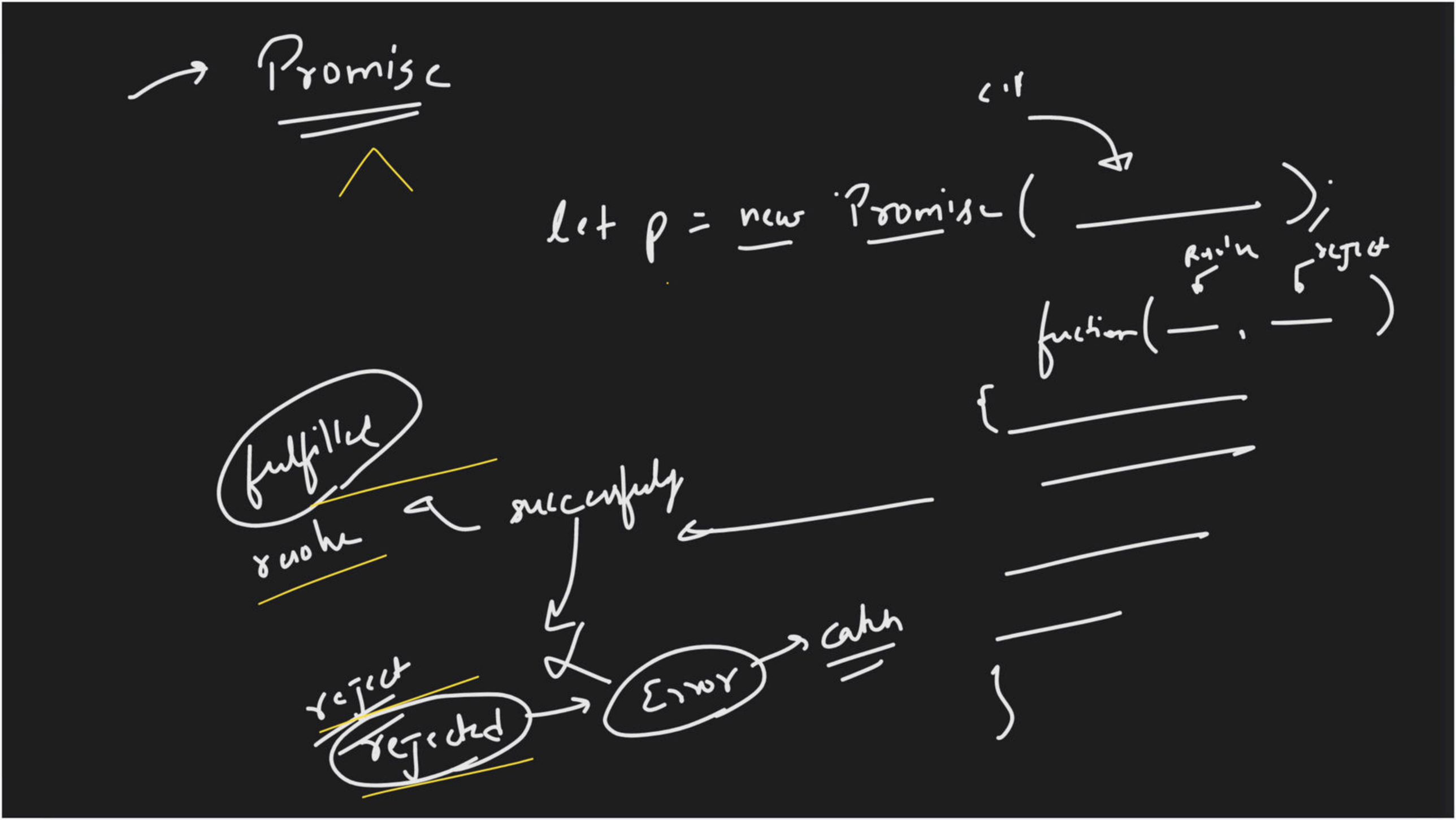
o
retrieve
send

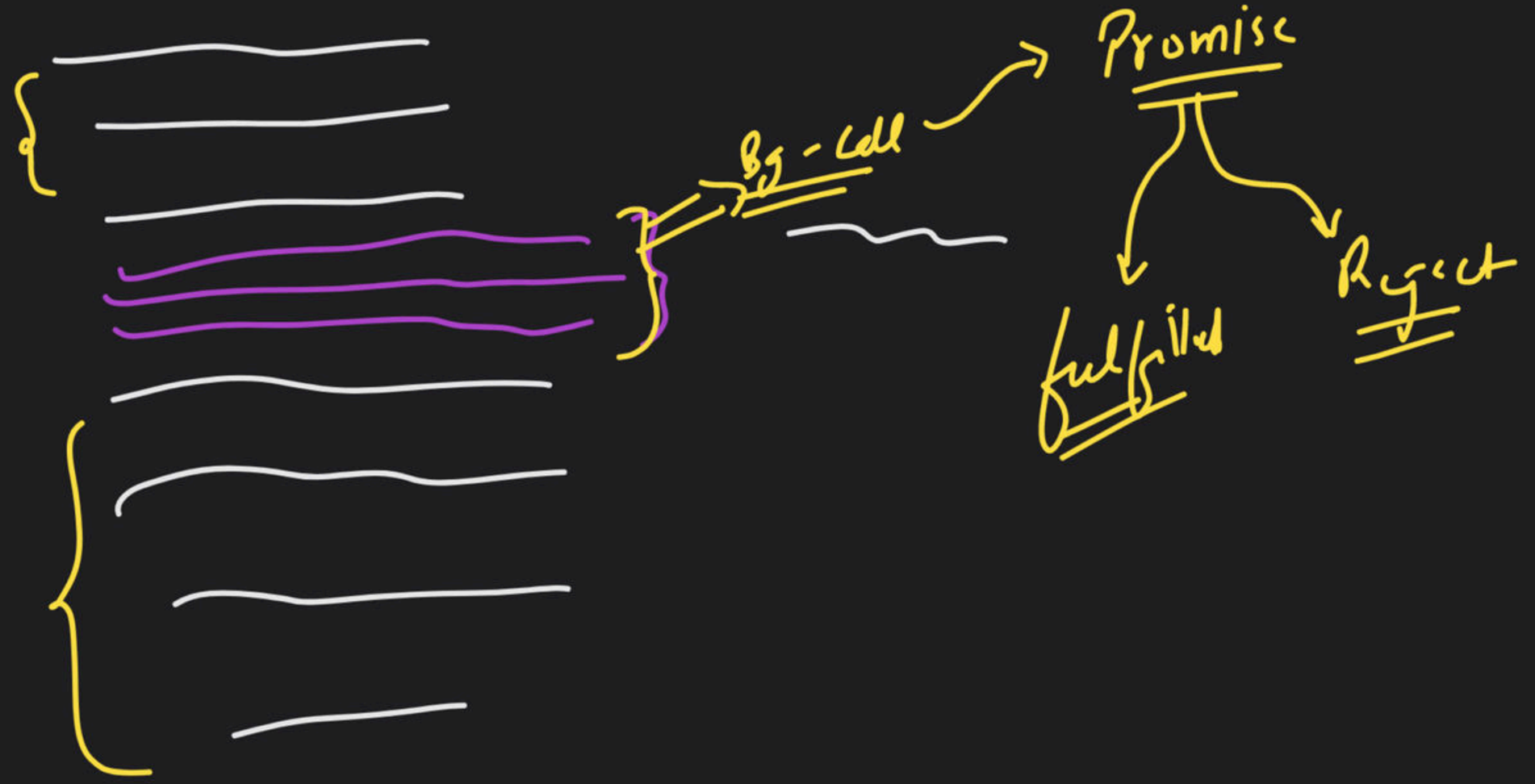
(get/post/put/
delete)

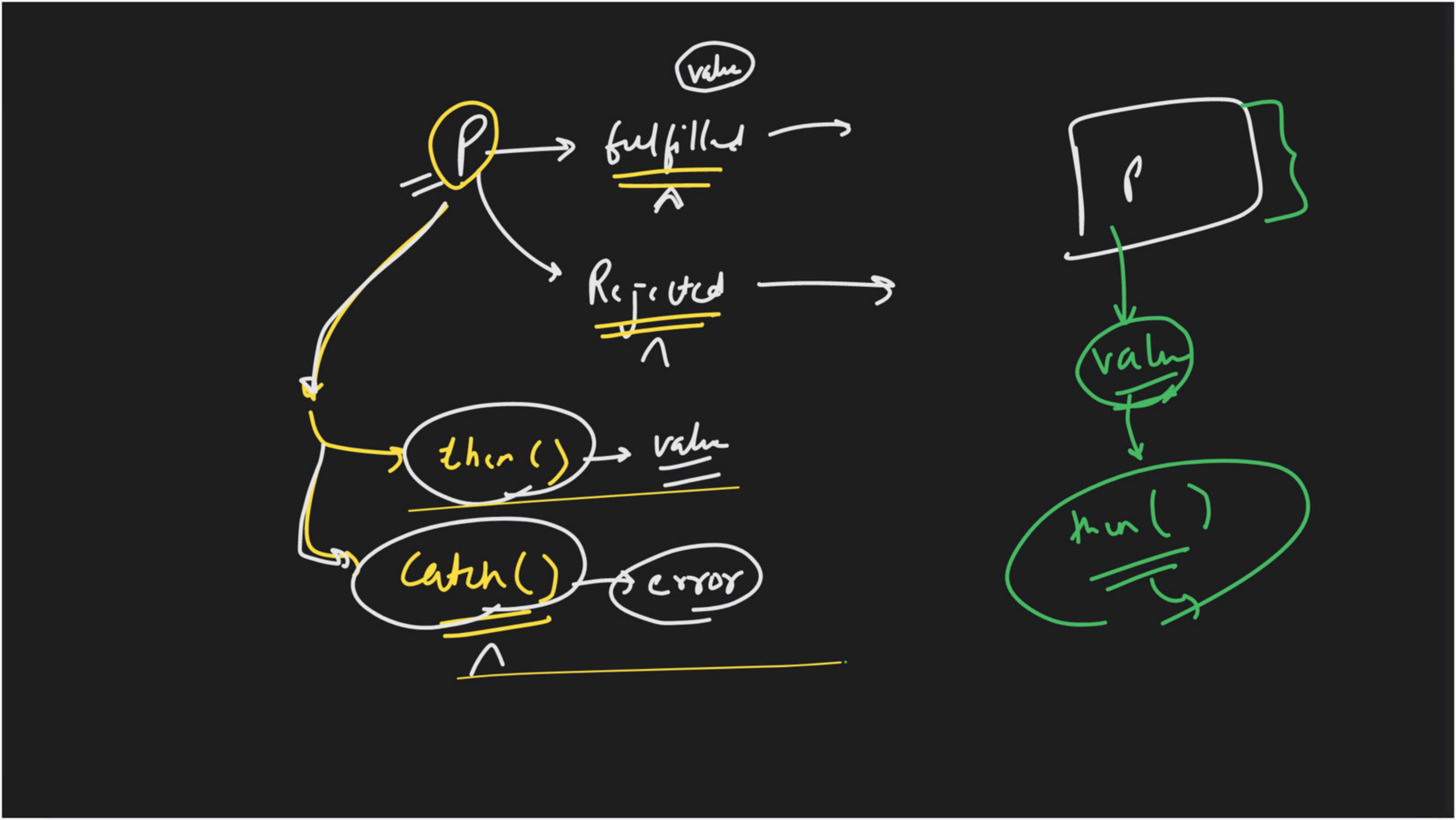
2 min

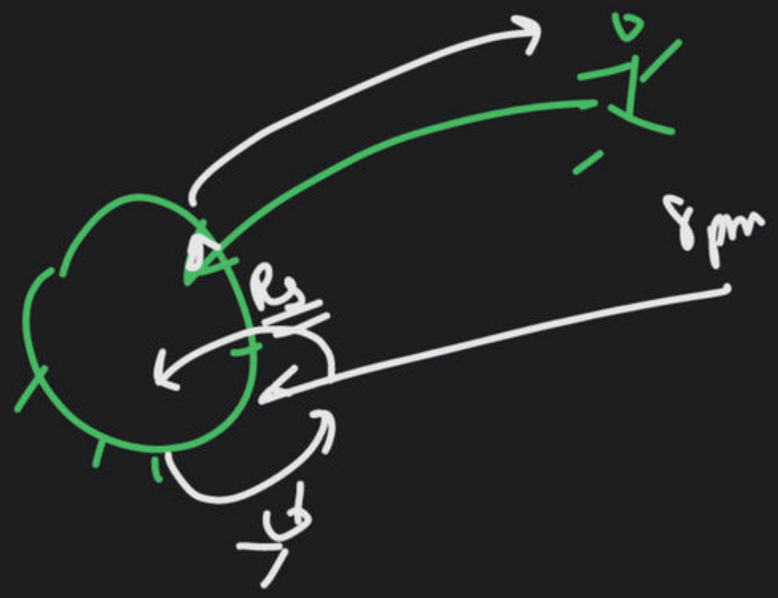
features of Async Code

- → clean & concise
- → better error handling
- → Easier debugging
- → Find Out → H/W

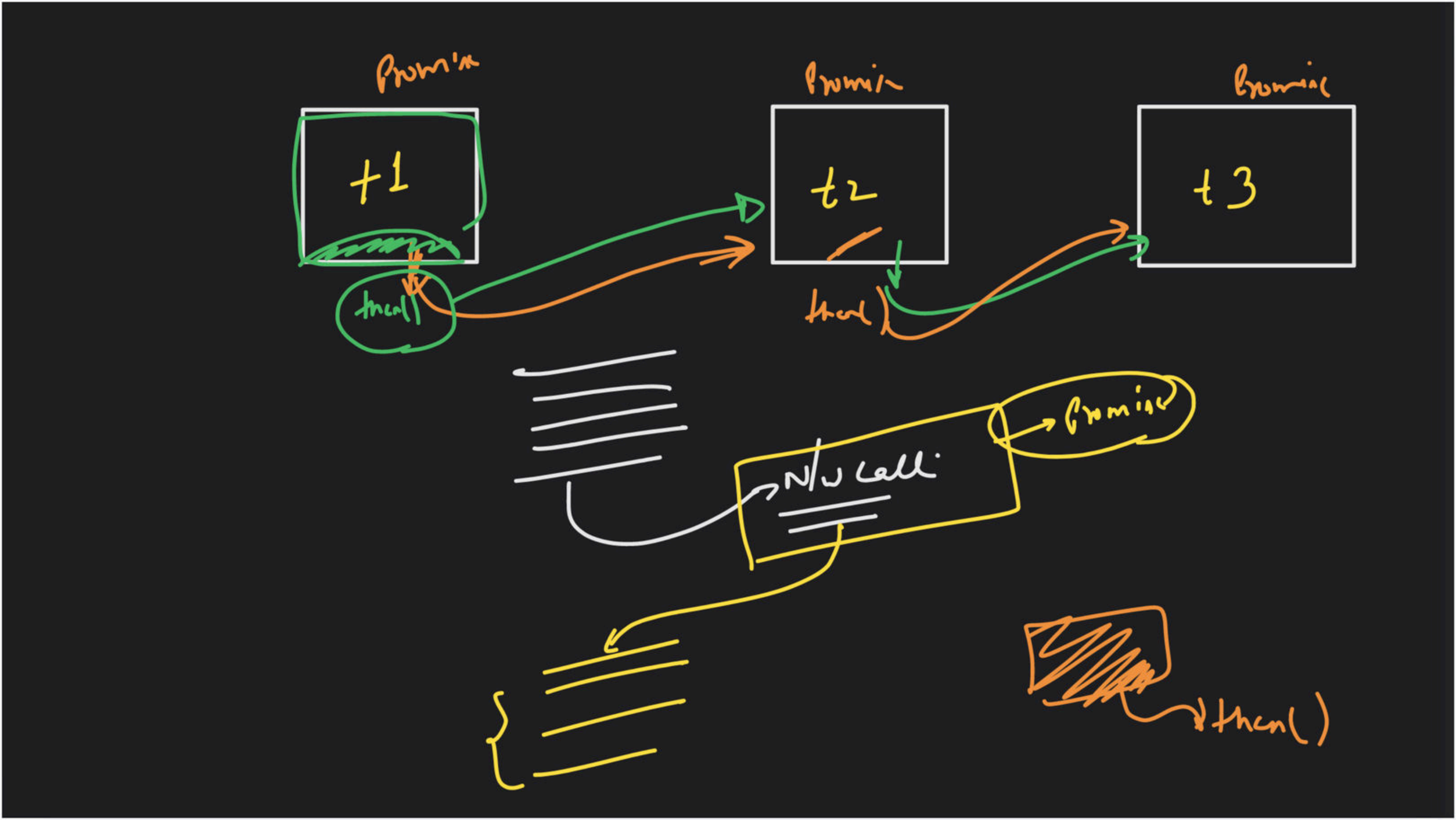

Promise
let p = new Promise( );
fulfilled
resolve
successfully
reject
rejected
Error
catch
function(,)
{
}

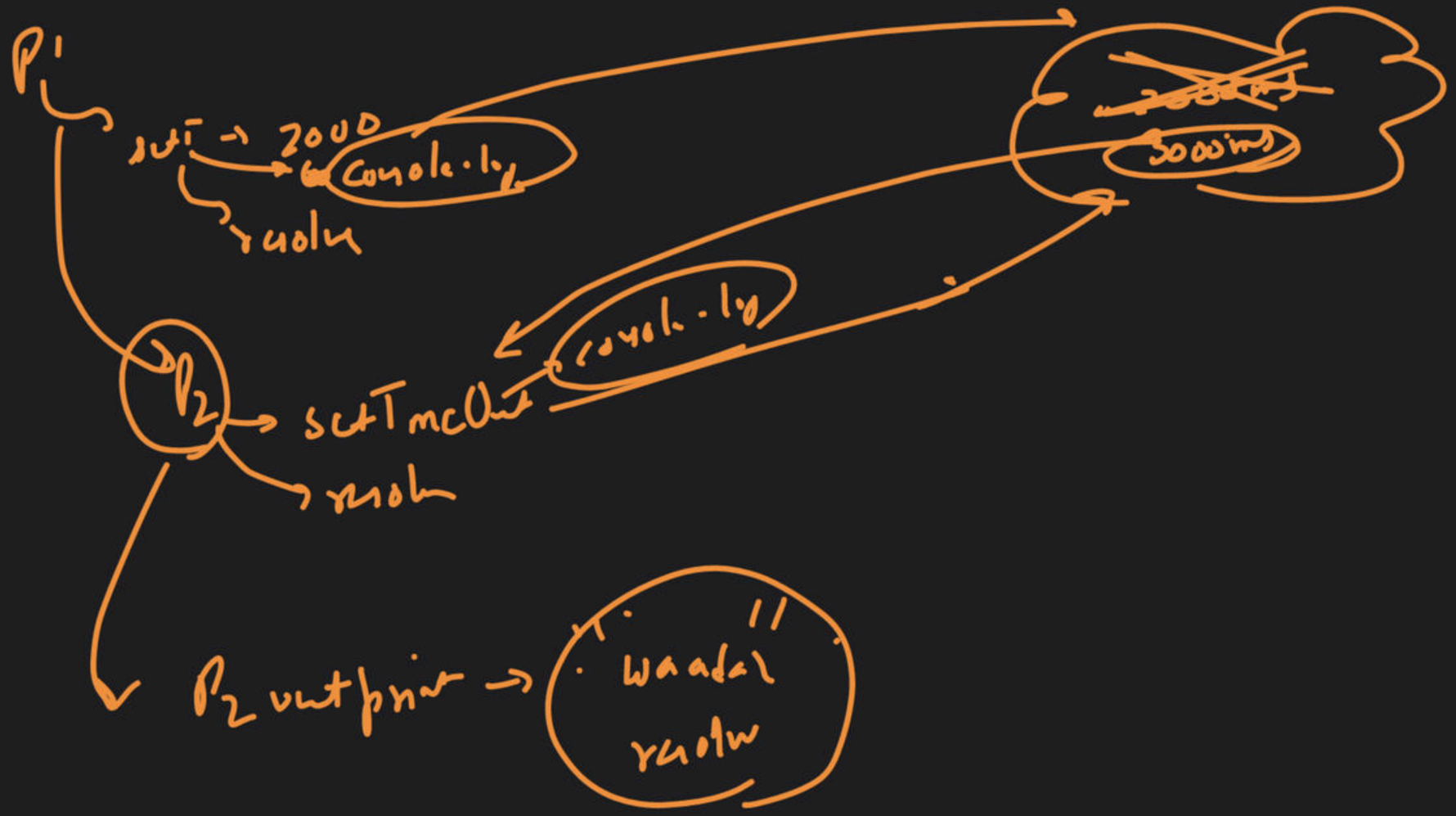

Promise
fulfilled

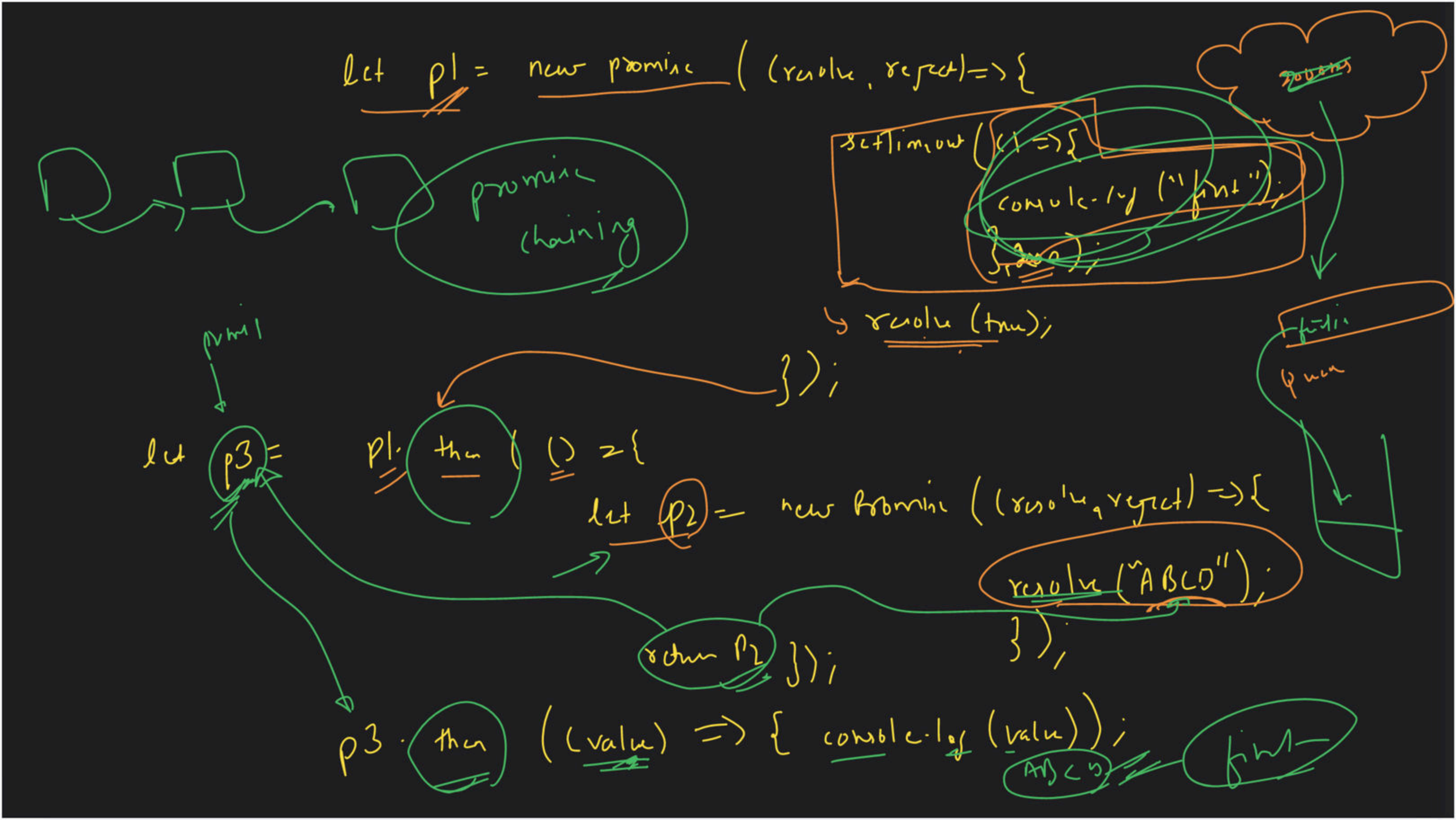

value
p
fulfilled
Rejected
then()
value
catch()
error
p
value
then()



then ( )


Promise
Promise
Promise
t1
t2
t3
then()
then()
N/J call
Promise
then()


P1
setT → 2000
console.log
resolve
P2
setTmcOut
console.log
resolve
P2 outprint →
waadal
resolve
5000ms


let p1 = new promise ((resolve, reject) => {
promise chaining
setTimeout (() => {
console.log ("first");
}, 2000);
resolve (true);
});
let p3 =
p1. then (() => {
let p2 = new Promise ((resolve, reject) => {
resolve ("ABCD");
});
return p2 });
p3 . then ((value) => { console.log (value) });
ABCD

So promises

So then ?

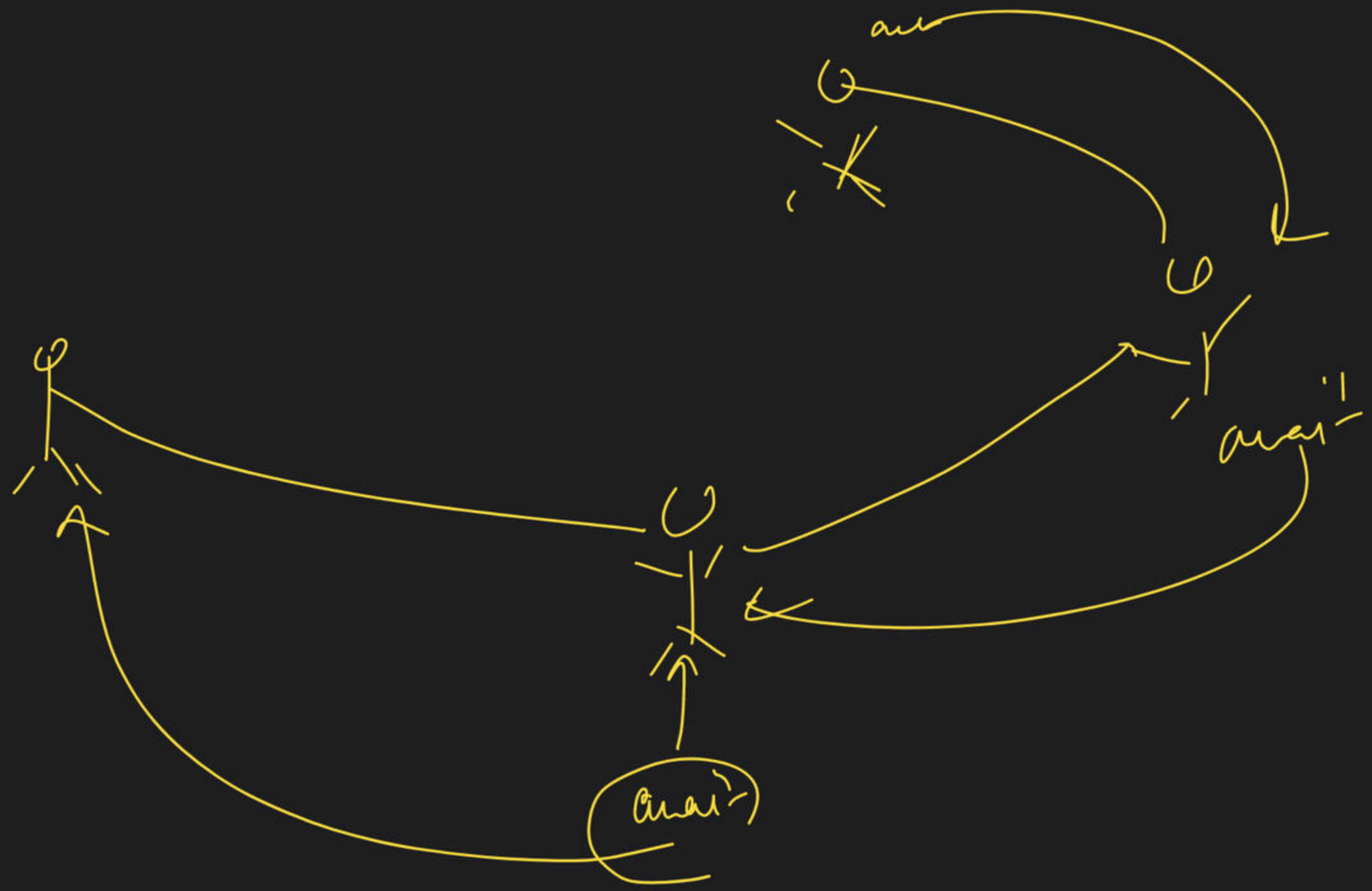
await p1

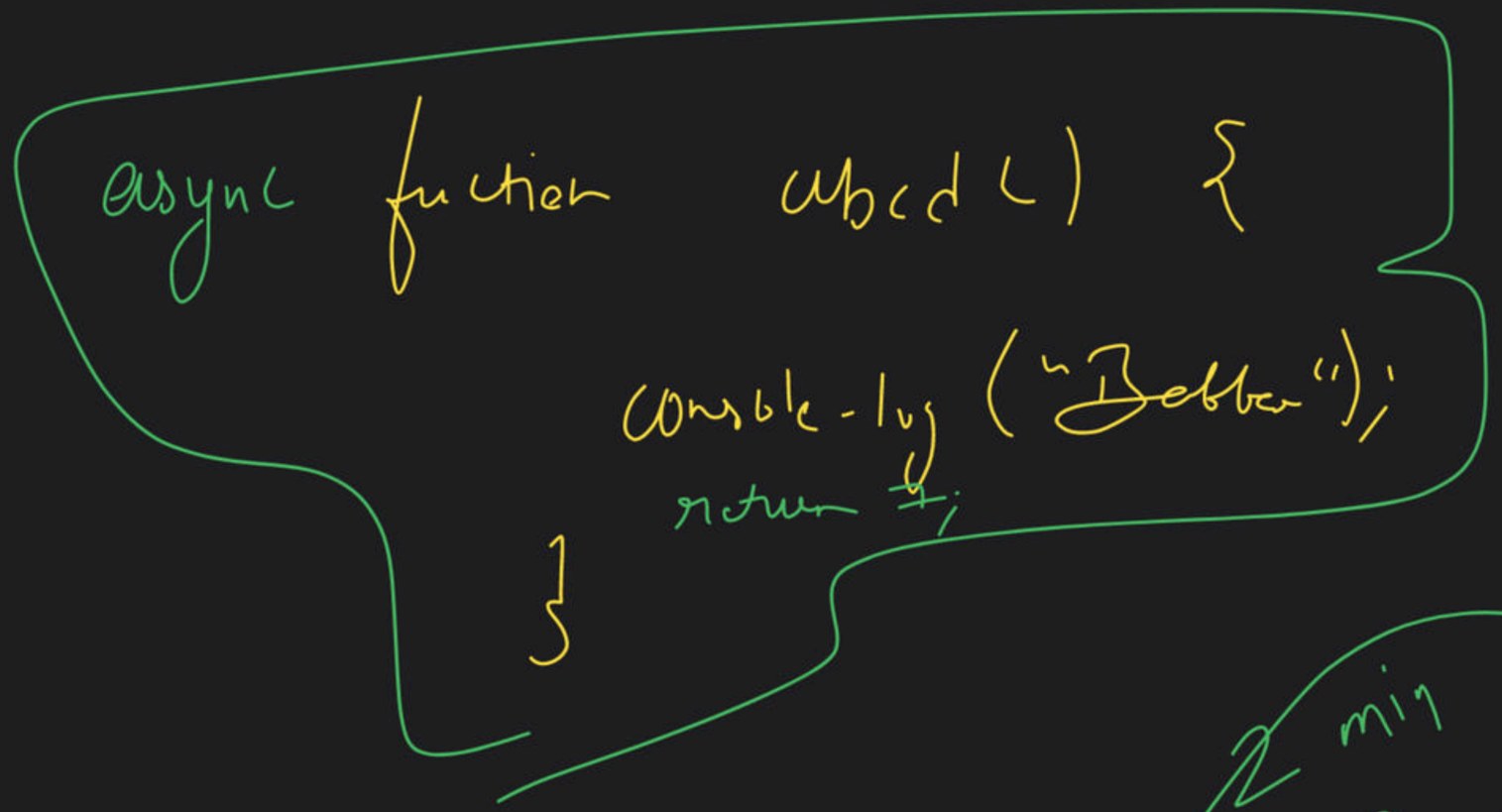
Async - await

special syntax used to work with promises

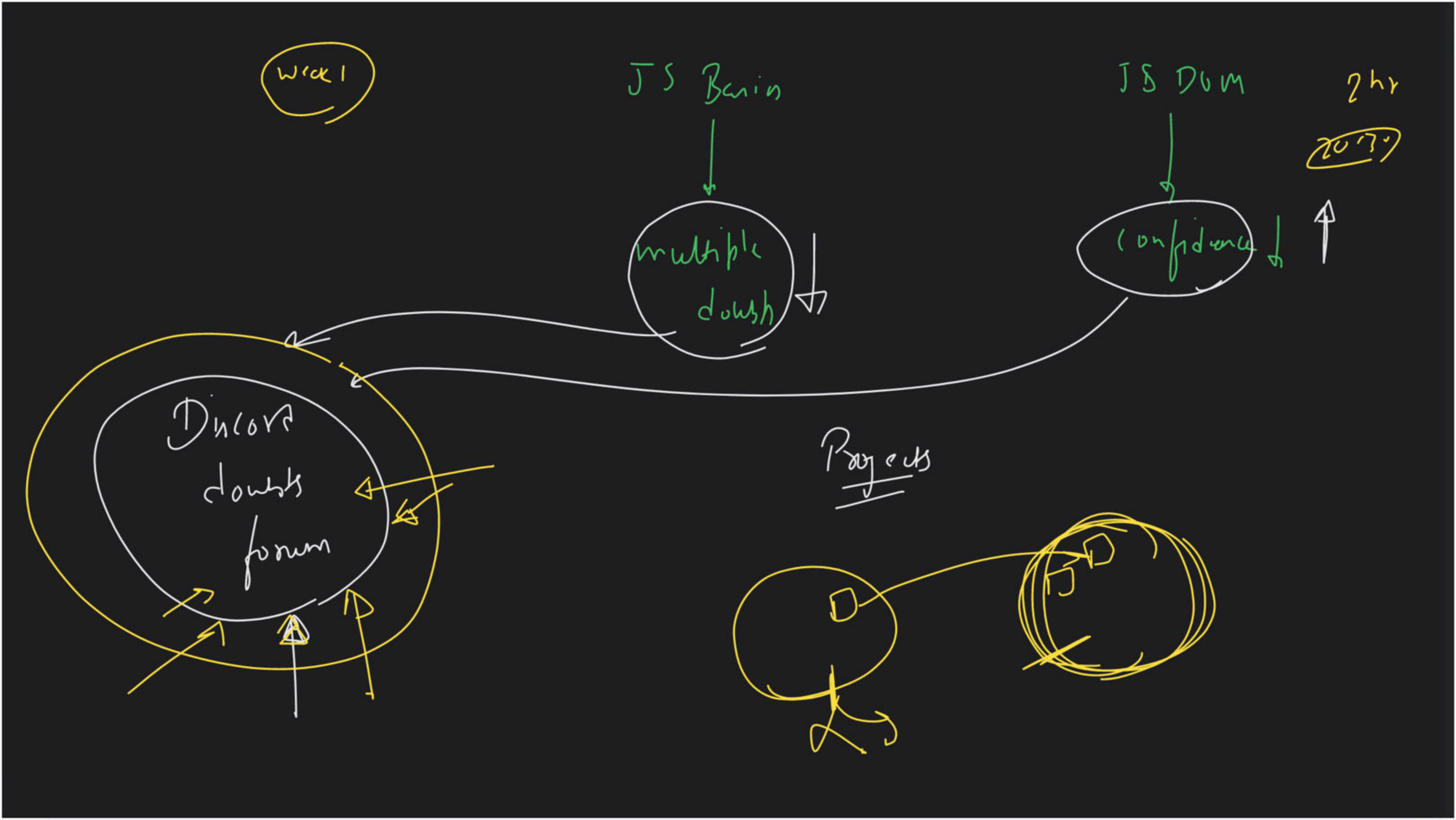
rukwana

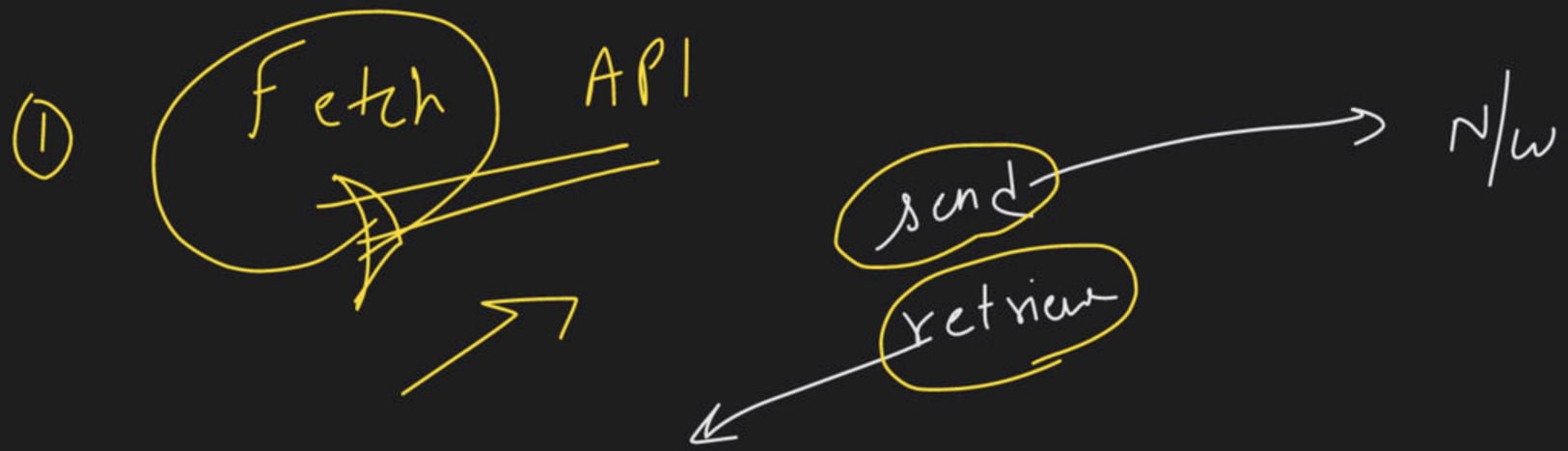
wait Kerne



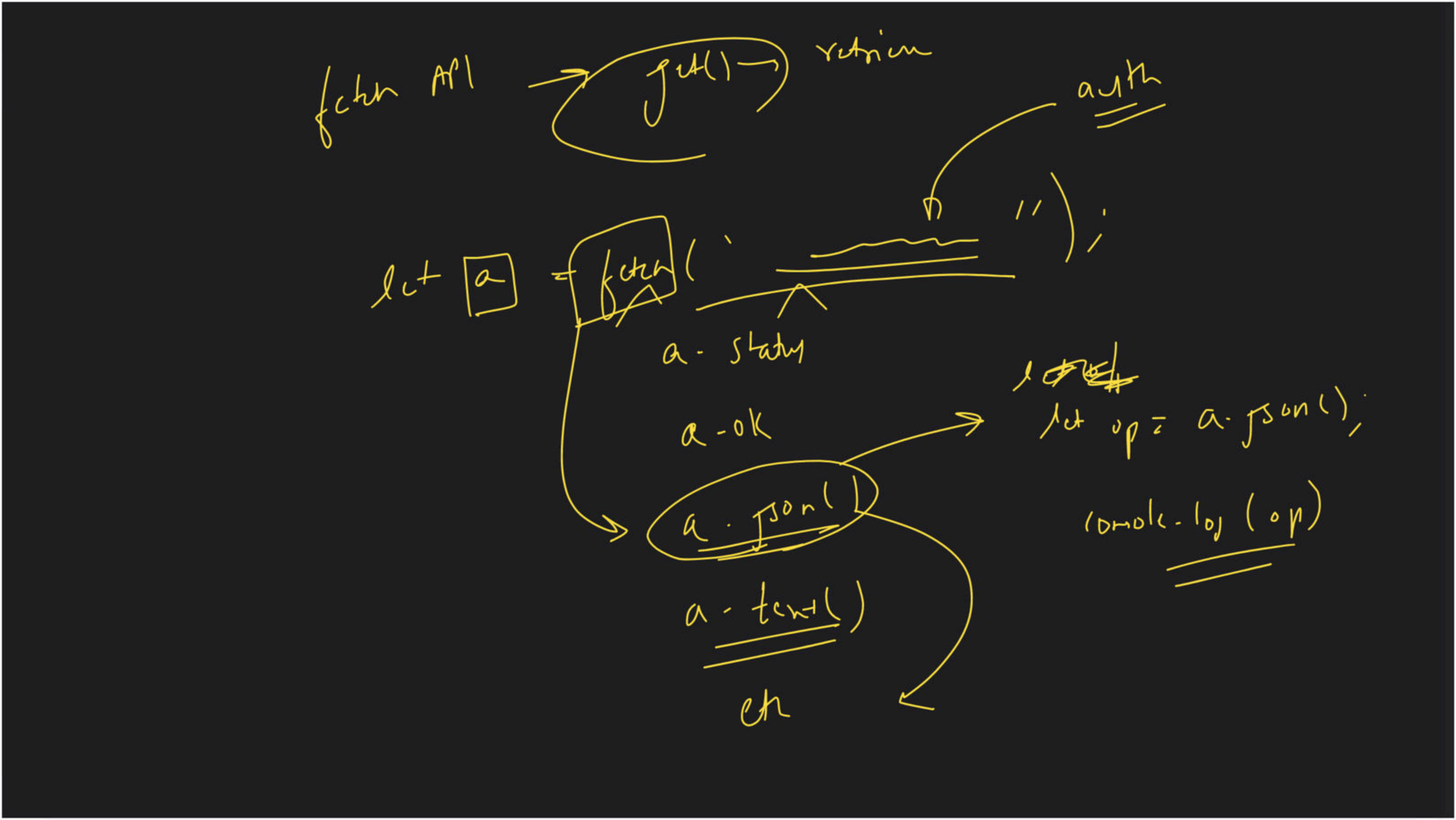

async function abcd() {
console.log("Bobby");
return 1;
}

2 min
Break


week 1
JS Brain
JS DOM
2hr
2023
multiple doubts
confidence
Discord doubts forum
Projects

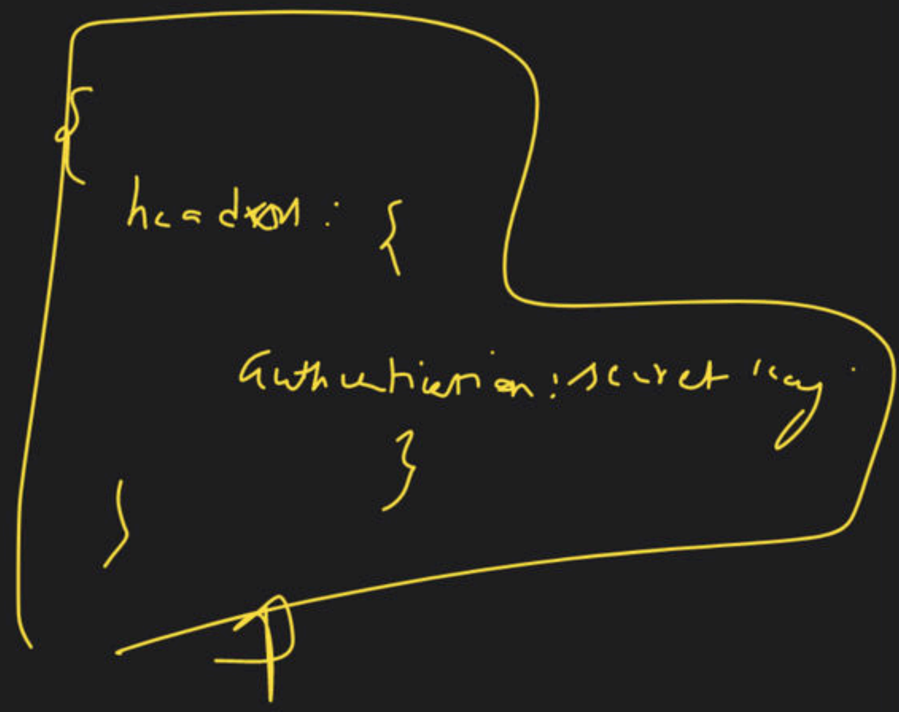


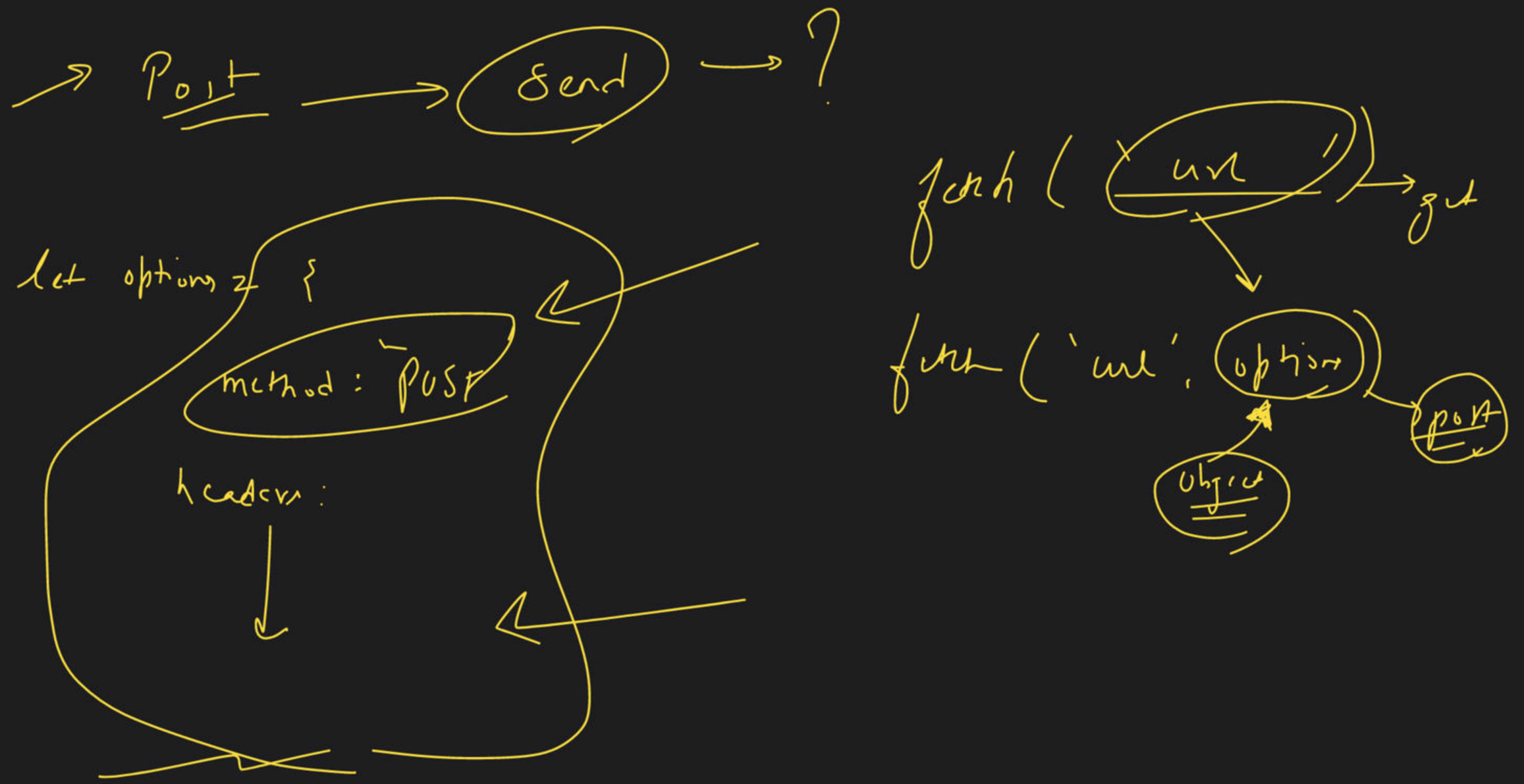
free APIs

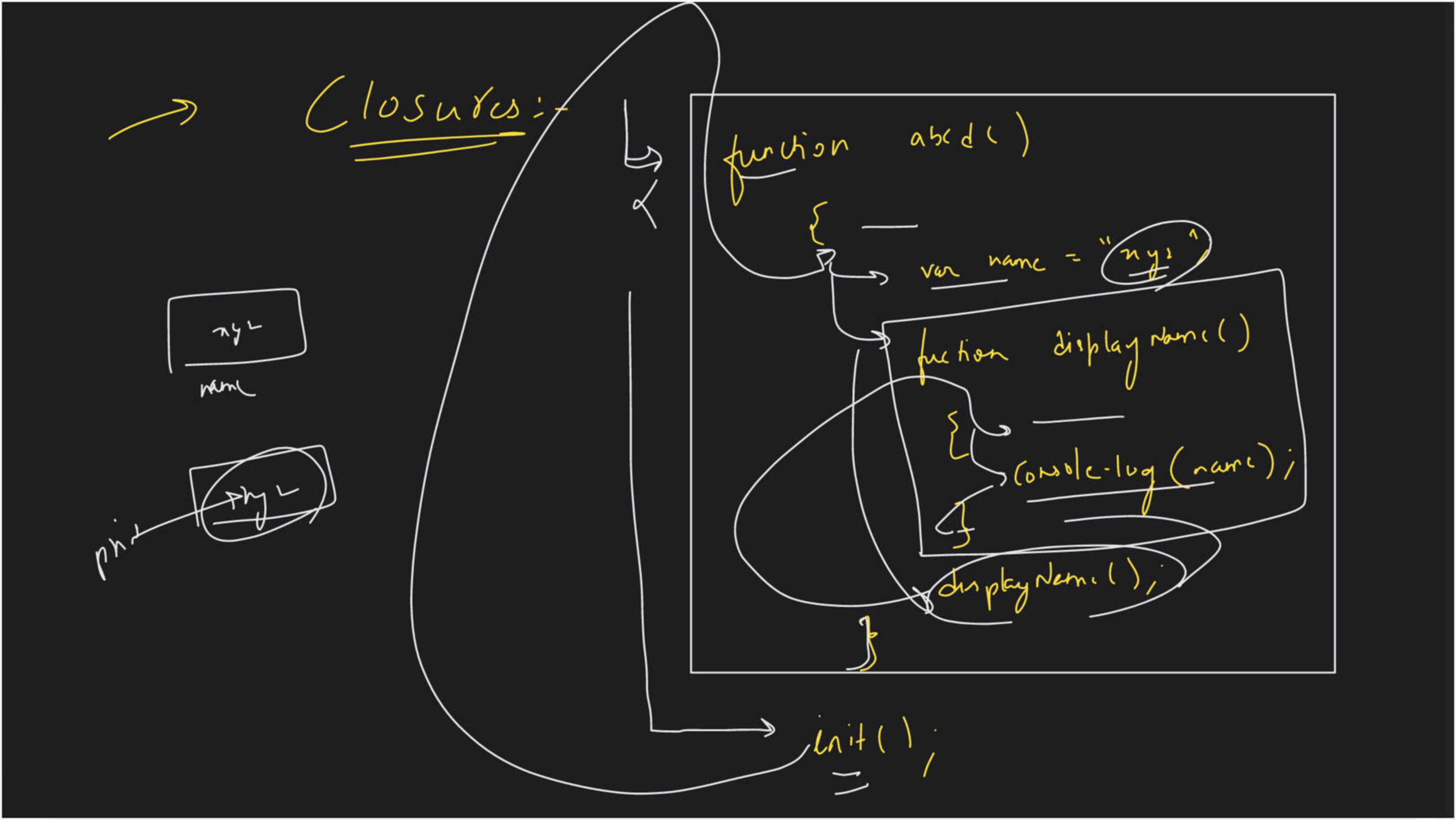

fetch API
get()
return
auth
let a = fetch(
a.status
a.ok
a.json()
a.text()
ch
let op = a.json();
console.log(op)

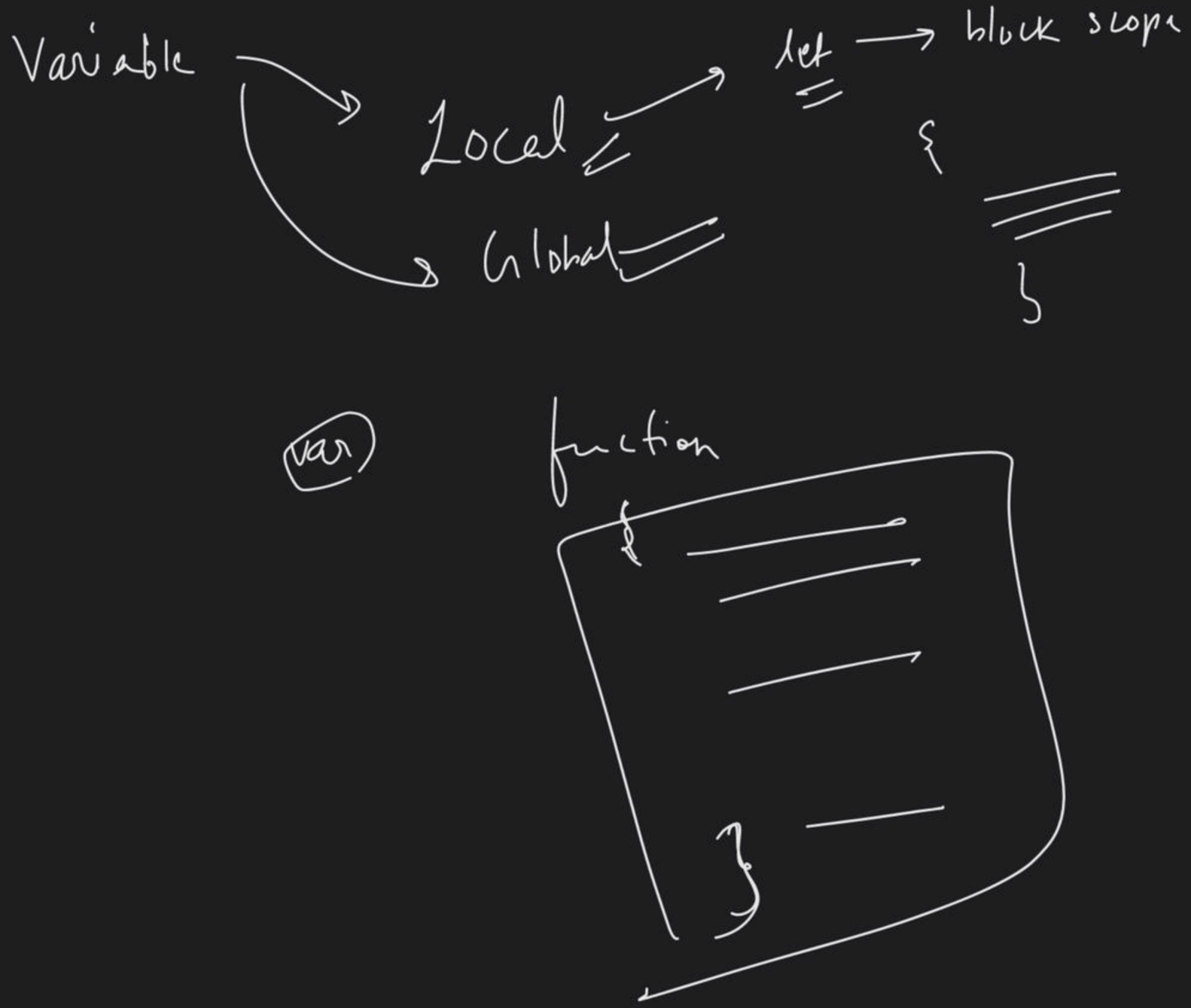
fetch (`_____` , [Options])

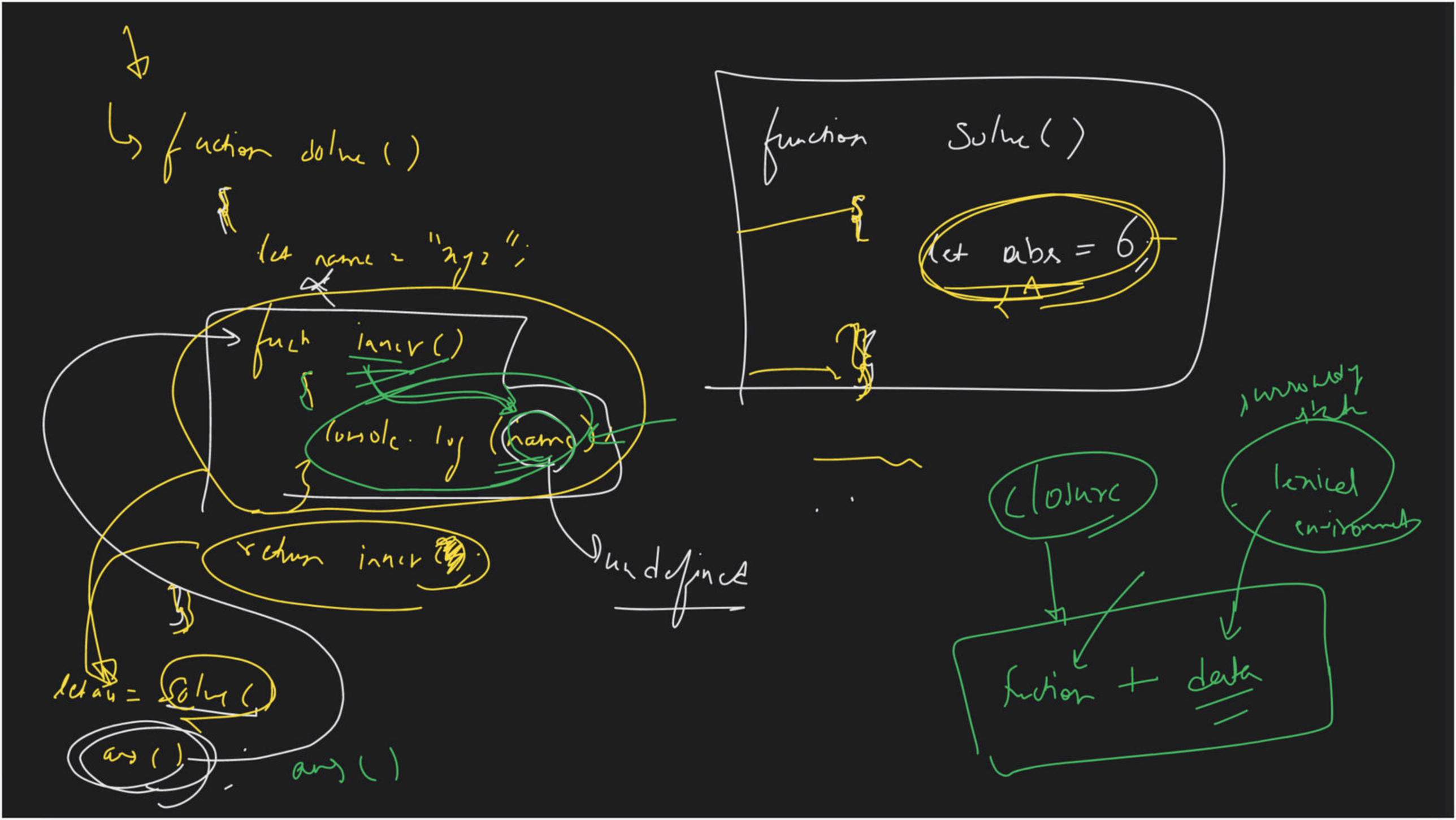
fetch (url, {
  headers : {
    authentication : secret key
  }
}



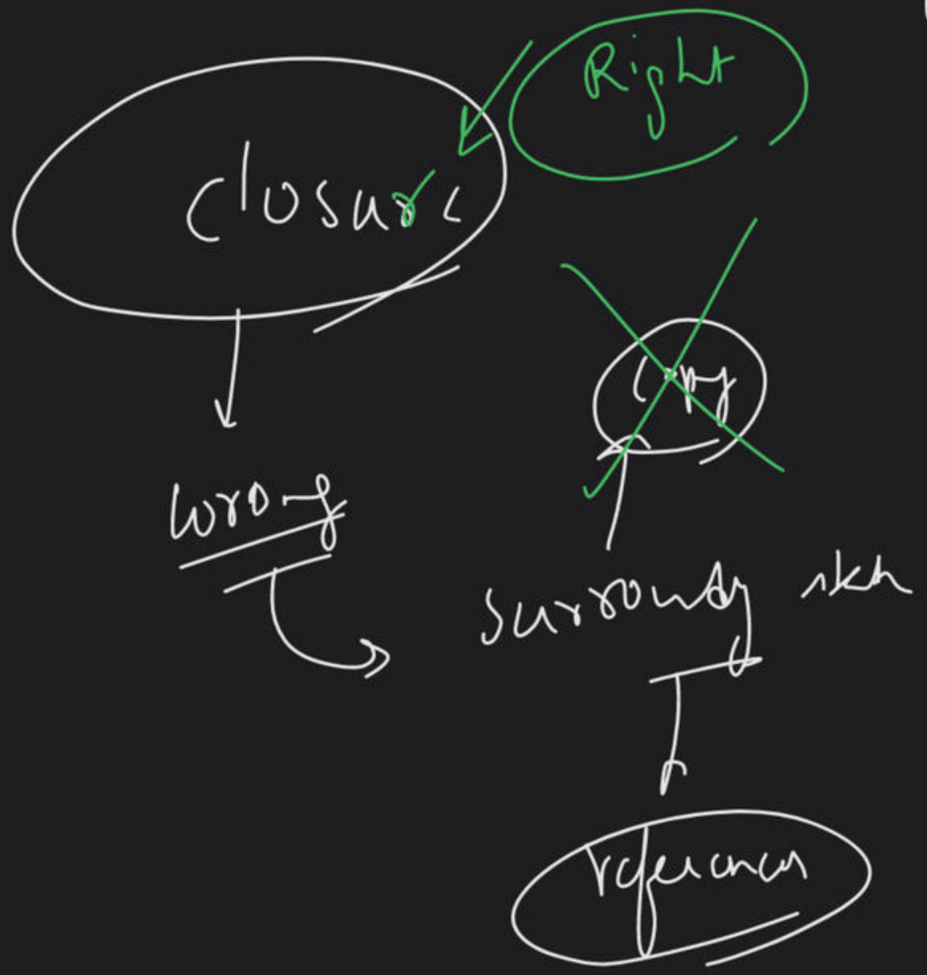

Post
Send
?
let options = {
method: 'POST'
headers:
fetch('url')
fetch('url', options)
port
object


Closures :-
function abcd()
{
var name = "xyz"
function displayName()
{
console.log(name);
}
displayName();
}
abcd();
xyz
name
phi
xyz

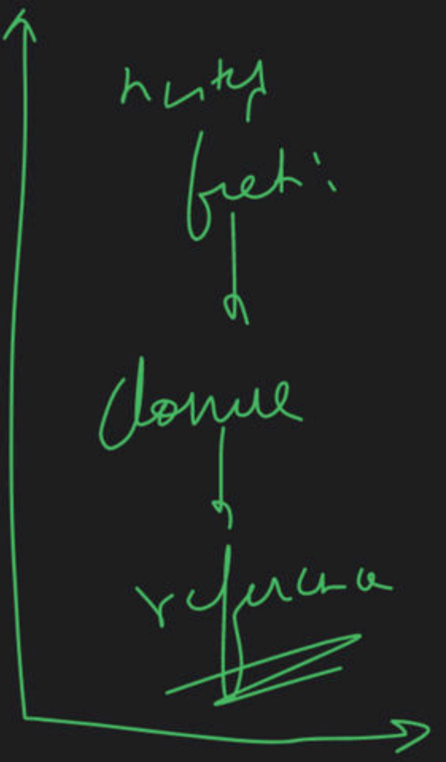

Variable
Local
let
block scope
{
}
Global
var
function
{
}

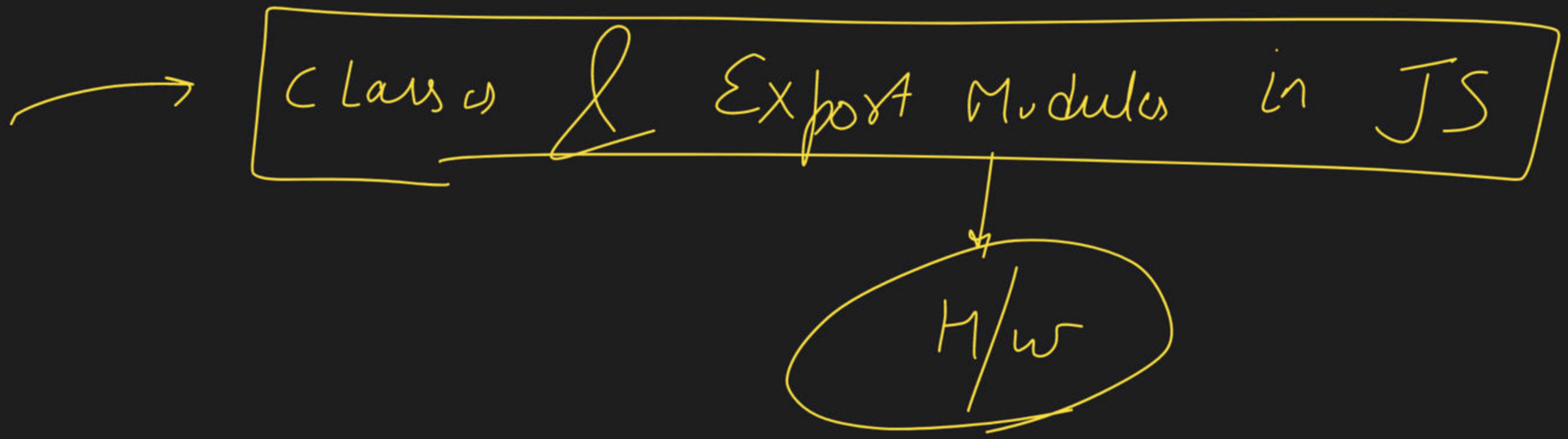

function doIt()
let name = "xyz";
function inner()
console.log(name)
return inner;
undefined
let a4 = doIt();
a4()
a4()
function sum()
let abx = 6;
closure
lexical environment
function + data

```
}
return inner
};
let ans = outer()
ans()
```




```
function outer()
{
    let a = 5;
    inner1()
    {
        let a = 3;
        console.log(a);
        inner2()
        {
            let a = 7;
            console.log(a);
            inner3()
            {
                console.log(a);
            }
        }
    }
}
```




Classes & Export Modules in JS
H/W