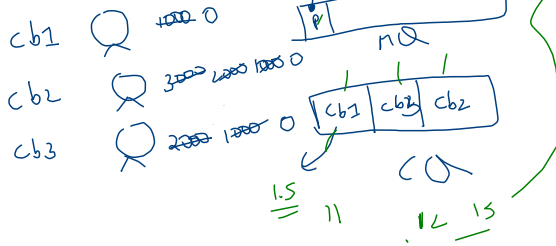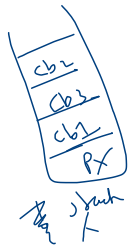Browser APIs

```
//cb+promise
console.log("start");

setTimeout(() => {
  console.log("setTimeout");
});

Promise.resolve().then(() => {
  console.log("resolve");
});

console.log("end");
```

cb queue
→ low priority
cb asyn

Cb
loop

आगे आकी
P asy

stack

event-loop

P_1 P_2 P_3 P_4 P_5
starved

cb

P_1 P_2 P_3
microtask queue

C_1 C_3 C_5 C_6 C_8
Callback queue

Console
start
end
resolve
setTimeout

high priority
micro

all
allset
race
any
resolve
reject

L
M
H

P − − −

ST (cb1, 1000) → 1sec ✗
after 1 sec

ST (cb2, 3000)

ST (Cb3, 2000)

P 0.5 / 1sec
P

cb1 🔵 1000 0
cb2 🔵 3000 2000 1000 0
cb3 🔵 2000 1000 0

nQ

Cb1 Cb3 Cb2

1.5   11   12 13

cb2
cb3
cb1
P✗

Async → return me a promise

async fonction foo(){

3

Await

await keyword makes JS wait until promise settles and return val