Promise → a promise basically represents upcoming completion or a failure of an asyc event. & it's resulting value.

---

* State | Value

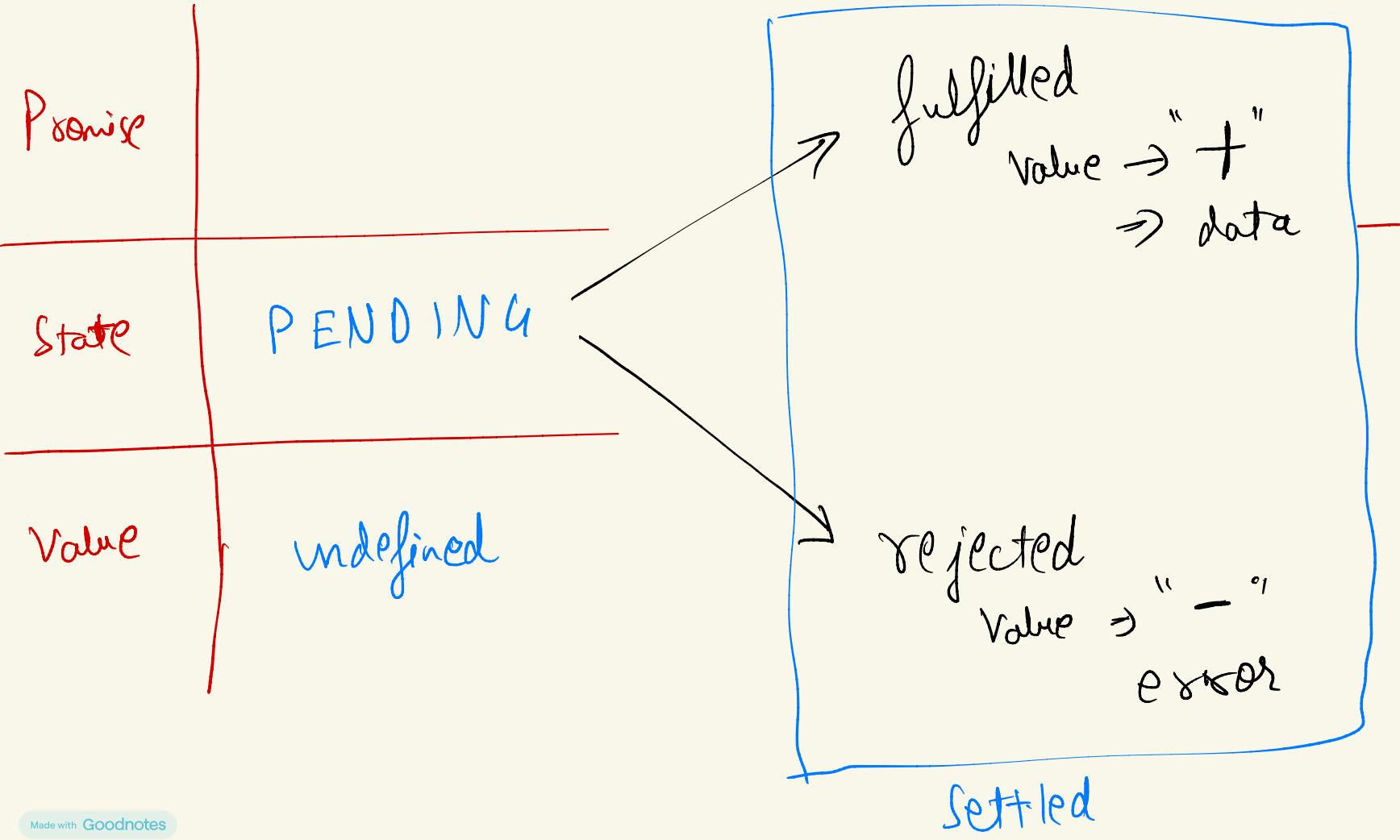
* Pending (State) & Value ⇒ undefined

(fulfilled) value ⇒ Playstation

boards % > 90 ⇒ Play Station

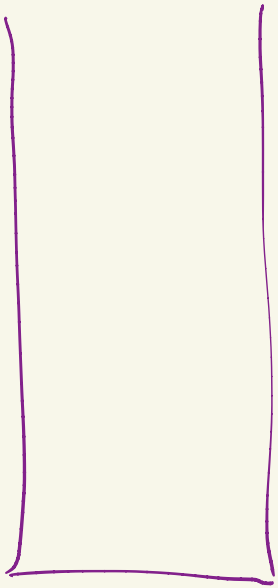
< 90 ⇒ - Personal Computer

failed value



```
const p1 = new Promise((res, rej) => {
    const condition = true;
    if (condition) res("here the promis is resolved");
    else rej("promise failed");
})
```

```
p1.then((data) => console.log(data))
  .catch((err) => console.log(err))
```



res & rej → Some piece of code

Sync

async

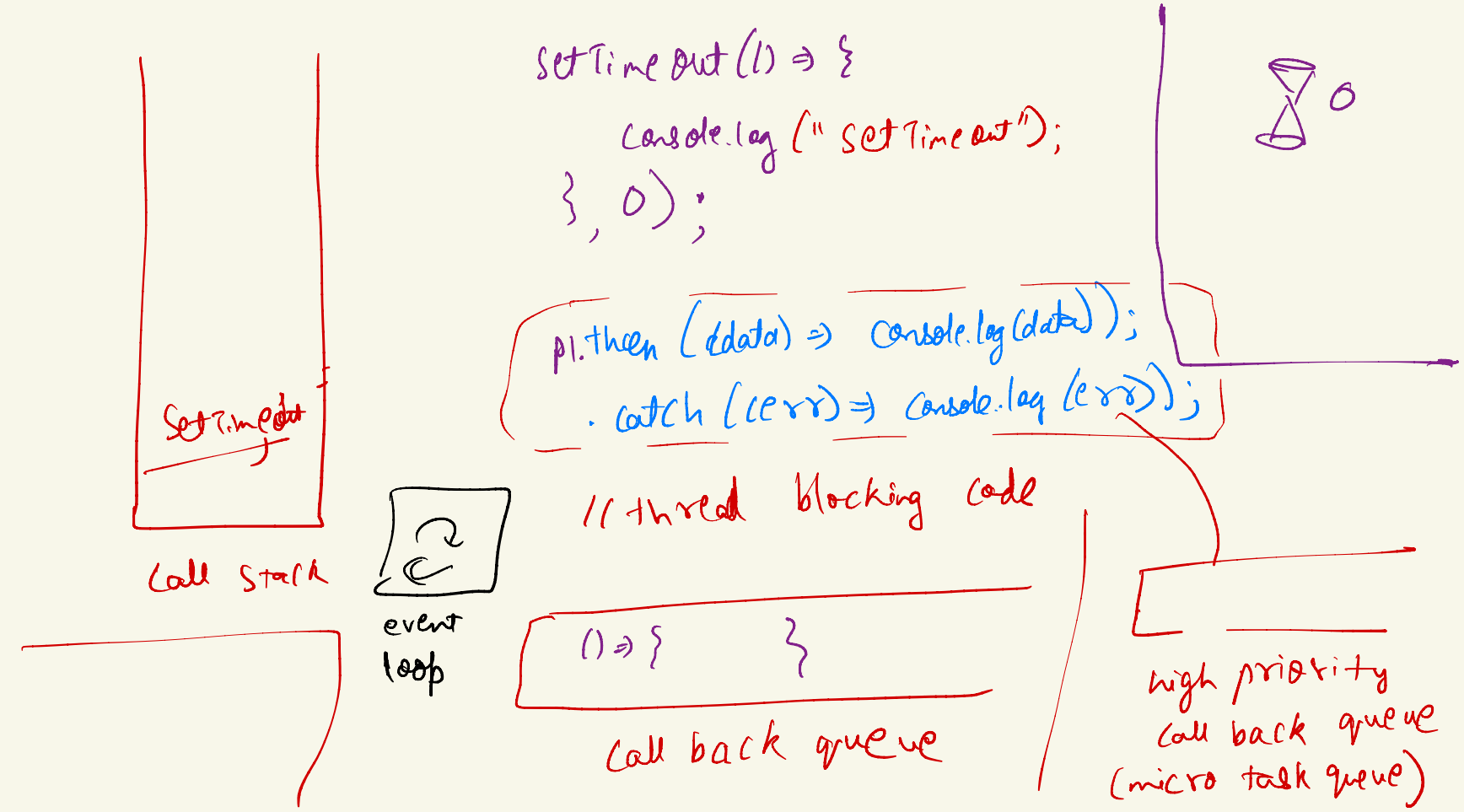
```
for(let i=0; i<10000; i++){
    console.log("some Sync");
}
```

→ Sync

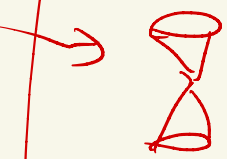
① → "Some Sync code" 10,000 times
→ resolved value / rejected
} ⇒ async

---

② → resolved value / rejected value
→ "Some Syn Code", 10,000 times
⇒ Sync



```
7   const p2 = new Promise((res, rej) => {
8     console.log("Hi I am inside promise");
9     const condtion = true;
10    if (condtion) {
11      setTimeout(() => {
12        res("resolved");
13      }, 1000);
14    } else rej("rejected");
15  });
16
17  for(let i=0;i<100000;i++){
18      console.log('some sync code');
19  }
20
21  p2.then((data) => console.log(data)).cat
```



() => { res("resolved") }

Call Stack

"Hi I am inside promise"
"some sync code"
"resolve"

res("resolve");