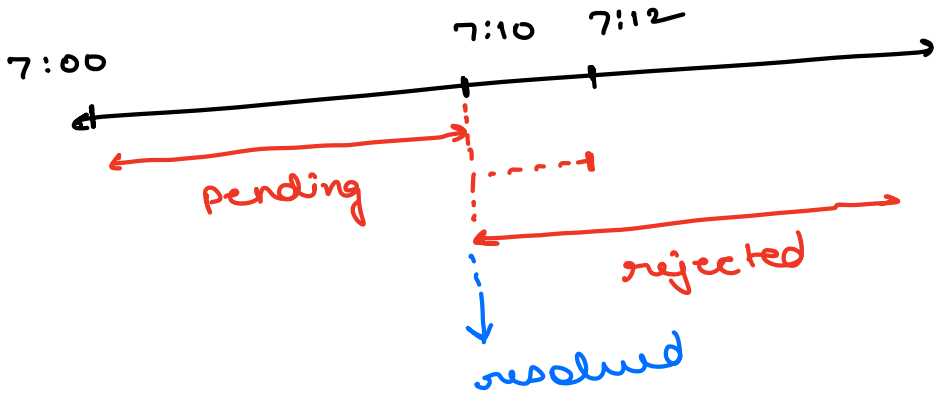
Agenda

→ What are Promise
→ Async programming with Promise
→ Chaining of Promise
→ Serial operation with promise

Promise:- In JS, a promise is an object representing the eventual completion or failure of an asynchronous task.

Note:- It provides a way to handle async tasks more elegantly while maintaining the code readability.



3 states

1. Pending
2. Fulfilled or resolved } Settled
3. Rejected

clean the room ~~success~~ ~~failed~~ → ice cream (Resolved)
→ no ice cream (rejected)

There are 3 methods that can be used to do something or react when the promise is settled (resolved or rejected)

→ .then (resolved)
→ .catch (rejected)
→ .finally

```
myPromise
  .then(function (val) {
    console.log("Success:", val);
  })
  .catch(function (val) {
    console.log("Failed:", val);
    return "testing";
  })      You, 43 seconds ago • Uncommit
  .then(function (val) {
    console.log(val);
  })
  .finally(function () {
    console.log("Accept your fate!");
  });
```

1) Make API call
2) Read DB
3) Write DB
4) Make API call

Call Stack

Web/Node APIs

```
fs.promises.readFile("./files/f1.txt", "utf-8");
fs.promises.readFile("./files/f2.txt", "utf-8");
fs.promises.readFile("./files/f3.txt", "utf-8");
```


```
const onFulfilled = (data) => {
  console.log("This is file data:", data);
};

const onRejected = (error) => {
  console.log("This is the error:", error);
};
```