The Complete JS Guide - then & catch











Promises in JavaScript

In the last part, we learned about promises and their different states

Syntax

```
let promise = new Promise(function (resolve, reject) {
   // your code
});
```



Resolve

---> We use resolve(value) — if the promise is finished successfully, with result value

Error

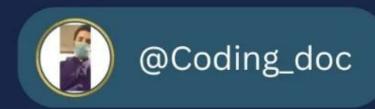
---> We use reject(error) — if an error has occurred, the error is the error object.



Example

let's create a new promise to store our value either in resolve or reject depending on the condition

```
let promise = new Promise((resolve, reject) => {
    let value = true
    if(value) {
        resolve("lahore")
    }else {
        reject("Karachi")
    }
}
```



then and catch

In JavaScript , a Promise has 2 consumers

---> then : when the Promise is successful, we use then

---> catch : when the Promise fails , we use catch



Case 1 (true condition)



Case 2 (false condition)

```
let promise = new Promise((resolve, reject) => {
    let value = false;
    if (value) {
        resolve("lahore");
    } else {
        reject("Karachi");
    }
};

promise.then((value) => console.log(value)).catch((err) => console.log(err));
// karachi
```



Points to note

```
let promise = new Promise((resolve, reject) => {
    let value = false;
    if (value) {
        resolve("lahore");
    } else {
        reject("Karachi");
    }
};

promise.then((value) => console.log(value)).catch((err) => console.log(err));
// karachi
// karachi
```

This will get the value of resolve

This will get the value of reject

