

Promises

27 January 2025 10:46

- ✳ A promise is a special java Script object that represents the eventual completion(or failure) of an asynchronous operation.

Three states:

- ✳ Pending: the initial state, the promise is still waiting for operation to finish.
- ✳ Fulfilled: the operation completed successfully.
- ✳ Rejected: the operation failed.

Explanation:

To create a promise **new promise()**.

The promise constructor takes a function with two parameters:

--> resolve: to mark as promise is successful

--> reject: to mark is as failed.

Handling():

.then() method is called when the promise is fulfilled.

It receives the results from the resolve().

.catch() method is called when the promise is rejected.

It receives the results from the reject().

Example:

```
<script>
  let myPromise =new Promise((resolve,reject)=>{
    let success=true
    if(success){
      resolve("the operation is successful")
    }
    else{
      reject("the operation is failed")
    }
  });
  myPromise
    .then(result=>{
      document.write(result)
    })
    .catch(error=>{
      document.write(error)
    });
</script>
```

Output:

the operation is successful

Chaining promises:

--> you can have chain multiple ,.then()

Ex:

```
<script>
  let promise=new Promise((resolve,reject)=>{
    resolve(5);
  });
  promise
    .then(result=>{
      document.write(result)
      return result*2
    })
    .then(result=>{
      document.write(result)
      return result+3;
    })
    .then(result=>{
      document.write(result)
    })
    .catch(error=>{
      document.write("error",error)
    })
</script>
```

Output:

5

10

13

Promise with set timeout:

```
<script>
let Promise = new Promise((resolve, reject) => {
    setTimeout(() => {
        resolve("Promise resolved after 2 seconds");
    }, 2000);
});

delayedPromise
    .then(result => {
        console.log(result);
    })
    .catch(error => {
        console.log("Error:", error);
    });
</script>
```

Output:

Promise resolved after 2 seconds.