

PROMISE IN JAVASCRIPT

In JavaScript, a promise is a good way to handle asynchronous operations. It is used to find out if the asynchronous operation is successfully completed or not.

A promise may have one of three states.

- pending
- fulfilled
- rejected

Creating a Promise

To create a promise object, we use the Promise() constructor.

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

The Promise() constructor takes a function as an argument. The function also accepts two functions resolve() and reject().

If the the promise returns successfully, the resolve() function is called. And, if an error occurs, the reject() function is called.

Using a Promise

- **then()** method : The then() method is used with the callback when the promise is successfully fulfilled or resolved.
- **catch()** method : The catch() method is used with the callback when the promise is rejected or if an error occurs.
- **finally()** method : The finally method() gets executed when the promise is either resolved successfully or rejected.

Using async / await

```
function fetchData(){  
  
    return new Promise((resolve,reject)=>{  
  
        setTimeout (()=>{resolve("Data fetched successfully")},2000))  
  
    }  
}
```

```
fetchData().then ((result)=>{ console.log(result);})
```

```
async function abc(){
```

```
    const result = await fetchData()
```

```
    console.log(result);
```

```
}
```

```
abc()
```