What is a promise in JavaScript?

The Promise in JavaScript may look quite complicated to understand at first sight, but in reality, it is quite simple and is not rocket science. In JavaScript, a promise is just like a promise that you make in real life to show that you are committed to doing something.

For example, I promise to get good marks in mathematics, and then this Promise has two outcomes, either it will be fulfilled (or resolved) or not fulfilled (or be rejected). So if I get good marks, the Promise is resolved, but if I fail to get good marks, it will not be resolved because I could not keep my Promise.

However, in [JavaScript](https://www.javatpoint.com/javascript-tutorial), a promise has three outcomes; Promise gets resolved, gets rejected, or pending state, which means the Promise is not completed yet but may get completed after some time, so it is not rejected till now and is in the pending state.

**Let's see another example to understand the concept of Promise in JavaScript:**

Suppose, we have made a request to fetch some data from the server by using the Promise if we get that data from the server successfully, the Promise will be considered as resolved successfully (or completed), but in case, we don't get that data from the server due to any reason, that means the Promise was rejected or not completed.

**Syntax of Promise in JavaScript**

1. Let p= new Promise((resolve,rejected)) =**>**{
2. Instructions
3. }

Let's see **how to create and use a promise in JavaScript?**

First of all, we have to create a promise using the constructor:

1. constPromise =newPromise

The Promise has two parameters as we already have seen the syntax of the Promise:

1. constPromise =newPromise((resolve, reject)=**>**{
2. Condition
3. });

Here is the last part that is the condition. If the condition is satisfied, the Promise will be resolved; otherwise, Promise will be rejected.

1. constPromise =newPromise((resolve, reject)=**>**{
2. let condition;
4. if(condition is met){
5. resolve('Promise is resolved successfully.');
6. }else{
7. reject('Promise is rejected');
8. }
9. });

So here is our first Promise. Now let's use it.

**The then() method**

As we already discussed above, for a promise, there are two main cases: one for the resolved and another for the rejection. When the Promise gets resolved, something will happen next that depends on what we want to do with the resolved Promise.

1. myPromise.then();

This ".then();" method is only called when the Promise is resolved( completed successfully), and what we passed in it will be displayed. For example, we can pass a message in it for the user.

1. myPromise.then((message)=**>**{
2. console.log(message);
3. });

**The .catch(); method**

This method is called when the promise is rejected (or failed). And we can also pass a message in it for the user. We can write the catch method just after the "then ()" method.

1. myPromise.then((message)=**>**{
2. console.log(message);
3. }).catch((message)=**>**{
4. console.log(message);
5. });

So if the Promise gets resolved, then we will get the message passed in the ".then()" method on the screen, but if the Promise gets rejected then, we will get the message we passed in the ".catch()" method. This means in both cases; we will get different messages on the screen.

**We can understand how promise works more easily with the help of the following example:**

**Program**

1. <!DOCTYPE html**>**
2. **<html** lang="en"**>**
3. **<head>**
4. **<body>**
5. **<script>**
6. const myPromise = new Promise((resolve, reject) =**>** {
7. let condition=9;
9. if(condition**<0**) {
10. resolve('Promise is resolved successfully and then() menthod will be called');
11. } else {
12. reject('Promise is rejected and catch() medhod will be called');
13. }
14. });
16. myPromise.then((message) =**>** {
17. console.log(message);
18. }).catch((message) =**>** {
19. console.log(message);
20. });
21. **</script>**
22. **</body>**
23. **</html>**

**Explanation of the program**

In the above program, we have created a promise using the constructor and passed two parameters resolve, reject. Inside the Promise definition, we also have created a variable "condition" and assigned the value 9 to it. In addition, using the (if) statement, we have checked the value of the variable. If the condition gets satisfied, the "if" part will execute and promise get resolved (the then() method will be called too), if the condition does not get satisfied, the else part will be executed, and Promise gets rejected(then catch() will be executed).

**Output**

To see the output in the console, press the "f12" key to open the inspect mode in the [web browser](https://www.javatpoint.com/browsers).

Note: You can modify the program code if you want to execute the else part of the Promise by changing the condition to true or false.