Promise

# Basic Example :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Promise Example 1</title>

</head>

<body>

    <h2>Basic Promise Example</h2>

    <p id="output">Waiting for the result...</p>

    <script>

        // Example 1: Basic Promise

        const myPromise = new Promise((resolve, reject) => {

            const success = true; // Change to false to test rejection

            if(success) {

                resolve("Promise was successful!");

            } else {

                reject("Promise failed!");

            }

        });

        // Handle the promise

        myPromise

            .then(result => {

                document.getElementById("output").innerText = result; // Display success message

            })

            .catch(error => {

                document.getElementById("output").innerText = error; // Display error message

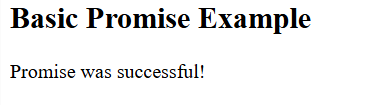
            });

    </script>

</body>

</html>

# OutPut:



# Promise with set time out :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Promise Example 2</title>

</head>

<body>

    <h2>Promise with setTimeout</h2>

    <p id="output">Fetching data...</p>

    <script>

        // Example 2: Promise with setTimeout to simulate async behavior

        function fetchData() {

            return new Promise((resolve, reject) => {

                setTimeout(() => {

                    const data = { user: "John Doe", age: 30 }; // Simulated data

                    resolve(data); // Resolve with data after 2 seconds

                }, 2000);

            });

        }

        // Call the fetchData function and handle the result

        fetchData()

            .then(result => {

                document.getElementById("output").innerText = `User: ${result.user}, Age: ${result.age}`;

            })

            .catch(error => {

                document.getElementById("output").innerText = `Error: ${error}`;

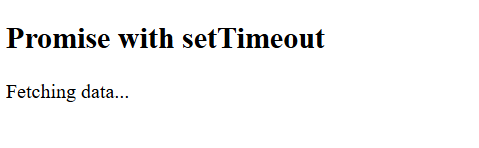
            });

    </script>

</body>

</html>

# OutPut :



# Chaining Multiple promise :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Promise Example 3</title>

</head>

<body>

    <h2>Chaining Multiple Promises</h2>

    <p id="output">Starting the process...</p>

    <script>

        // Example 3: Chaining Multiple Promises

        function step1() {

            return new Promise((resolve, reject) => {

                setTimeout(() => {

                    resolve("Step 1 complete");

                }, 1000);

            });

        }

        function step2() {

            return new Promise((resolve, reject) => {

                setTimeout(() => {

                    resolve("Step 2 complete");

                }, 1000);

            });

        }

        function step3() {

            return new Promise((resolve, reject) => {

                setTimeout(() => {

                    resolve("All complete");

                }, 1000);

            });

        }

        // Chain the promises

        step1()

            .then(result => {

                document.getElementById("output").innerText = result;

                return step2(); // Continue to step 2

            })

            .then(result => {

                document.getElementById("output").innerText = result;

                return step3(); // Continue to step 3

            })

            .then(result => {

                document.getElementById("output").innerText = result; // Final step complete

            })

            .catch(error => {

                document.getElementById("output").innerText = `Error: ${error}`;

            });

    </script>

</body>

</html>

# Output :

