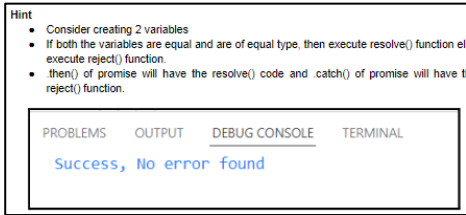


Problem statement:

With your knowledge gained on JS Promises consider handling an asynchronous JS operation.

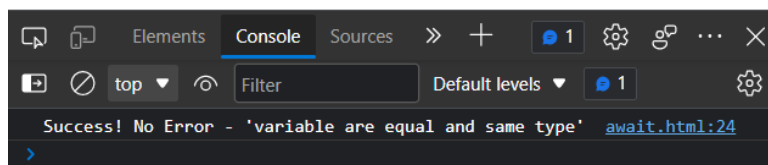


Solution:

Code (Resolve Case):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JS Async Await & Promise</title>
</head>
<body>
  <h3>Javascript Promise Resolve and Reject Error Handling</h3>
  <script>
    var promise = new Promise(function(resolve, reject){
      let a = 10;
      let b = 10;
      //checks condition if both variables are equal and are of equal type
      if(a === b){
        resolve();
      }else{
        reject();
      }
    });
    //resolve code
    promise.then(function(){
      console.log("Success! No Error - 'variable are equal and same type'");
    })
    //reject code
    .catch(function(){
      console.log("Error Occured - 'variables aren't same'");
    });
  </script>
</body>
</html>
```

Output (Resolve Case):



Code (Reject Case):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JS Async Await & Promise</title>
</head>
<body>
  <h3>Javascript Promise Resolve and Reject Error Handling</h3>
  <script>
    var promise = new Promise(function(resolve, reject){
      let a = 10;
      let b = "10";
      //checks condition if both variables are equal and are of equal type
      if(a === b){
        resolve();
      }else{
        reject();
      }
    });
    //resolve code
    promise.then(function(){
      console.log("Success! No Error - 'variable are equal and same type'");
    })
    //reject code
    .catch(function(){
      console.log("Error Occured - 'variables aren't same'");
    });
  </script>
</body>
</html>
```

Output (Reject Case):

