

```
function CompareTwoStrings(str1, str2) {  
  return new Promise((resolve, reject) => {  
    if (str1 === str2) resolve("Strings are equal");  
    else reject("Strings are not equal");  
  });  
}  
  
CompareTwoStrings("Hello", "hello")  
  .then((res1) => console.log(res1))  
  .catch((res2) => console.log(res2));
```

```
async function fun() {  
  return 1;  
}  
  
fun()  
  .then((res) => console.log(res))  
  .catch((err) => console.log(err));
```

```
async function display() {  
  let promiseToTest = new Promise((resolve, reject) => {  
    setTimeout(() => console.log("Done!"), 1000);  
  });  
  
  let result = await promiseToTest;  
  console.log(result);  
}  
  
display();
```

```
function calc(a, b) {  
  return new Promise(function (resolve, reject) {  
    setTimeout(function () {  
      resolve(a + b); //9  
    }, 1000);  
  });  
}
```

```
}
```

```
// promise chaining
```

```
calc(3, 6)  
  .then((res1) => res1 + 2)  
  .then((res2) => res2 + 3)  
  .then((res3) => res3 + 4)  
  .then((res3) => console.log(res3));
```

```
// async await
```

```
async function add() {  
  const res1 = await calc(3, 6);  
  const res2 = await calc(res1, 2);  
  const res3 = await calc(res2, 3);  
  return await calc(res3, 4);  
}
```

```
add().then((x) => console.log(x));
```

```
async function f() {  
  try {  
    let response = await fetch("http://no-such-url");  
    let user = await response.json();  
  } catch (err) {  
    console.log(err);  
  }  
}
```

```
f();
```

```
function findDigitSum(n) {  
  return new Promise(function (resolve, reject) {
```

```

    if (!isNaN(n)) {
      sum = 0;
      while (n > 0) {
        rem = n % 10;
        sum = sum + rem;
        n = Math.floor(n / 10);
      }
      resolve(sum);
    } else {
      reject("Invalid, Not a number!");
    }
  });
}

async function findResult(n) {
  try {
    var result = await findDigitSum(n);
    console.log(result);
  } catch (error) {
    console.log(error);
  }
}

findResult(134);

```

```

function findAbsolute(n) {
  return new Promise(function (resolve, reject) {
    if (!isNaN(Math.abs(n))) {

```

```
    resolve("Absolute value!!");  
  } else {  
    reject("Invalid, Not a Number!!");  
  }  
});  
}
```

```
async function findResult(n) {  
  try {  
    var result = await findAbsolute(n);  
    console.log(result);  
  } catch (error) {  
    console.log(error);  
  }  
}
```

```
findResult(-189);
```