



# Promise In JavaScript



[www.recursiveminds.com](http://www.recursiveminds.com)

# Promise



**JS** promise.js

```
const promise = new Promise((resolve, reject) => {
  // asynchronous operation
  if (operationWasSuccessful) {
    resolve(value);
  } else {
    reject(error);
  }
});
```

```
promise
  .then(value => {
    // handle success
  })
  .catch(error => {
    // handle error
  });

```

# Example - 1



JS promise.js

```
// Example of promise
const myPromise = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Recursive Minds");
  }, 2000);
});

myPromise
  .then(result => {
    console.log(result);
  })
  .catch(error => {
    console.error(error);
});
```



console

Recursive Minds

# Example - 2



JS promise.js

```
const fetchData = url => {
  return new Promise((resolve, reject) => {
    fetch(url)
      .then(response => {
        if (!response.ok) {
          reject(response.statusText);
        }
        return response.json();
      })
      .then(data => {
        resolve(data);
      })
      .catch(error => {
        reject(error);
      });
  });
};

fetchData("https://api.example.com/data")
  .then(data => {
    console.log(data);
  })
  .catch(error => {
    console.error(error);
  });
}
```

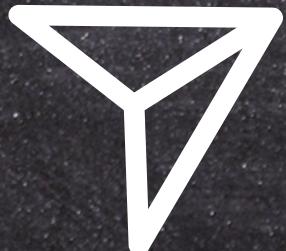
# Advantages over callbacks

1. Better handling of error
2. Improved readability
3. Better composability
4. Avoiding callback hell
5. Better debugging

# Read Detailed Article

visit:

[www.recursiveminds.com](http://www.recursiveminds.com)



[www.recursiveminds.com](http://www.recursiveminds.com)