

Promises allow us to turn our horizontal code (callback hell):

```
func1(function (value1) {  
  func2(value1, function (value2) {  
    func3(value2, function (value3) {  
      func4(value3, function (value4) {  
        func5(value4, function (value5) {  
          // Do something with value 5  
        });  
      });  
    });  
  });  
});
```

Into vertical code:

```
func1(value1)  
  .then(func2)  
  .then(func3)  
  .then(func4)  
  .then(func5, value5 => {  
    // Do something with value 5  
  });
```

Prior to ES6, we used bluebird or Q. Now we have Promises natively:

```
new Promise((resolve, reject) =>  
  reject(new Error('Failed to fulfill Promise'))  
  .catch(reason => console.log(reason)));
```

Where we have two handlers, **resolve** (a function called when the Promise is **fulfilled**) and **reject** (a function called when the Promise is **rejected**).

**Benefits of Promises:** Error Handling using a bunch of nested callbacks can get chaotic. Using Promises, we have a clear path to bubbling errors up and handling them appropriately. Moreover, the value of a Promise after it has been resolved/rejected is immutable - it will never change.

Here is a practical example of using Promises:

```
var request = require('request');  
  
return new Promise((resolve, reject) => {  
  request.get(url, (error, response, body) => {  
    if (body) {  
      resolve(JSON.parse(body));  
    } else {  
      resolve({});  
    }  
  });  
});
```

We can also **parallelize** Promises to handle an array of asynchronous operations by using `Promise.all()`:

```
let urls = [  
  '/api/commits',  
  '/api/issues/opened',  
  '/api/issues/assigned',  
  '/api/issues/completed',  
  '/api/issues/comments',  
  '/api/pullrequests'  
];  
  
let promises = urls.map((url) => {  
  return new Promise((resolve, reject) => {  
    $.ajax({ url: url })  
      .done((data) => {  
        resolve(data);  
      });  
  });  
});  
  
Promise.all(promises)  
  .then((results) => {  
    // Do something with results of all our promises  
  });
```