Creating Promises in ES6

creating promises and resolving/rejecting them using the 'resolve' and 'reject' keywords

Consider the following code:

```
let promise1 = new Promise( function( resolve, reject ) {
    // call resolve( value ) to resolve a promise
    // call reject( reason ) to reject a promise
} );

// Create a resolved promise
let promise2 = Promise.resolve( 5 );
console.log(promise1)
console.log(promise2)
```

When instantiating a promise, the handler function decides whether to resolve or reject the promise. When you call `resolve`, the promise moves to Fulfilled state. When you call `reject`, the promise moves to the Rejected state.

`Promise.resolve(value)` creates a promise that's already resolved.

Now, let's talk about promises that have either been fulfilled or rejected.