

Promises and Error Handling



Mark Zamoyta

SOFTWARE DEVELOPER AND EDUCATOR

@markzamoyta



Introduction



Errors in JavaScript

**Error Handling Using try and catch
finally**

Developer Defined Errors

Creating a Promise

Settling a Promise



Errors in JavaScript



Errors in JavaScript

```
let car = newCar;
```

```
console.log('continuing...');
```



Error Handling Using try and catch



try and catch

```
try {  
    let car = newCar;  
}  
catch(error) {  
    console.log('error: ', error);  
}  
console.log('continuing...');
```



finally



try, catch, finally

```
try {  
    let car = newCar;  
}  
catch(error) {  
    console.log('error: ', error);  
}  
finally {  
    console.log('this always executes');  
}
```



Developer Defined Errors



throw

```
try {  
    // code here...  
    throw new Error('my custom error');  
}  
catch(error) {  
    console.log('error: ', error);  
}  
finally {  
    console.log('this always executes');  
}
```



Creating a Promise



Create a Promise

```
let promise = new Promise(  
  function(resolve, reject) {  
    setTimeout(resolve, 100, 'someValue');  
  }  
);
```



Settling a Promise



Settling a Promise

```
let promise = new Promise(  
  function(resolve, reject) {  
    setTimeout(resolve, 100, 'someValue');  
  }  
);  
promise.then(  
  value => console.log('fulfilled: ' + value),  
  error => console.log('rejected: ' + error)  
);
```



Summary



Errors in JavaScript

- Stop execution?

Error Handling Using try and catch

- catch executes on an exception

finally

- always executes

Developer Defined Errors

- throw



Summary



Creating a Promise

- `function(resolve, reject)`

Settling a Promise

- `then(fulfilledFn, rejectedFn)`

