



**I bet you didn't know  
you could do this in a  
browser**

ANDREI ZVONIMIR CRNKOVIĆ



# Write better JS with Promises

KRISTIЈAN BAMBIR

**JavaScript is single threaded,  
yet highly asynchronous.**

# CALLBACK

```
function fetchData(url, callback) {  
    // ...  
    callback(data);  
}
```

```
fetchData('http://...', function(data) {  
    console.log('Received data:', data);  
});
```

## 🔥 CALLBACK HELL 🔥

```
fetchData(function(a) {  
  fetchMoreData(a, function(b) {  
    fetchMoreData(b, function(c) {  
      fetchMoreData(c, function(d) {  
        fetchMoreData(d, function(e) {  
          // ...  
        });  
      });  
    });  
  });  
});
```

**A Promise represents a value which may be available now, or in the future, or never.**

# BASICS

- Provides a simpler alternative for executing, composing, and managing asynchronous operations.
- Can be chained.
- Handles asynchronous errors using an approach similar to `try/catch`.

# CREATING A PROMISE

```
var promise = new Promise(function(resolve, reject) {  
  // do things, validate data...  
  
  if (isValid) {  
    resolve('It worked!');  
  } else {  
    reject(Error('It broke'));  
  }  
});
```



# CONSUMING A PROMISE

```
promise.then(function(result) {  
    console.log(result); // "It worked!"  
    return result;  
}).catch(function(err) {  
    console.log(err); // Error: "It broke"  
});
```

// or...

```
promise.then(function(result) {  
    console.log(result); // "It worked!"  
    return result;  
}, function(err) {  
    console.log(err); // Error: "It broke"  
});
```

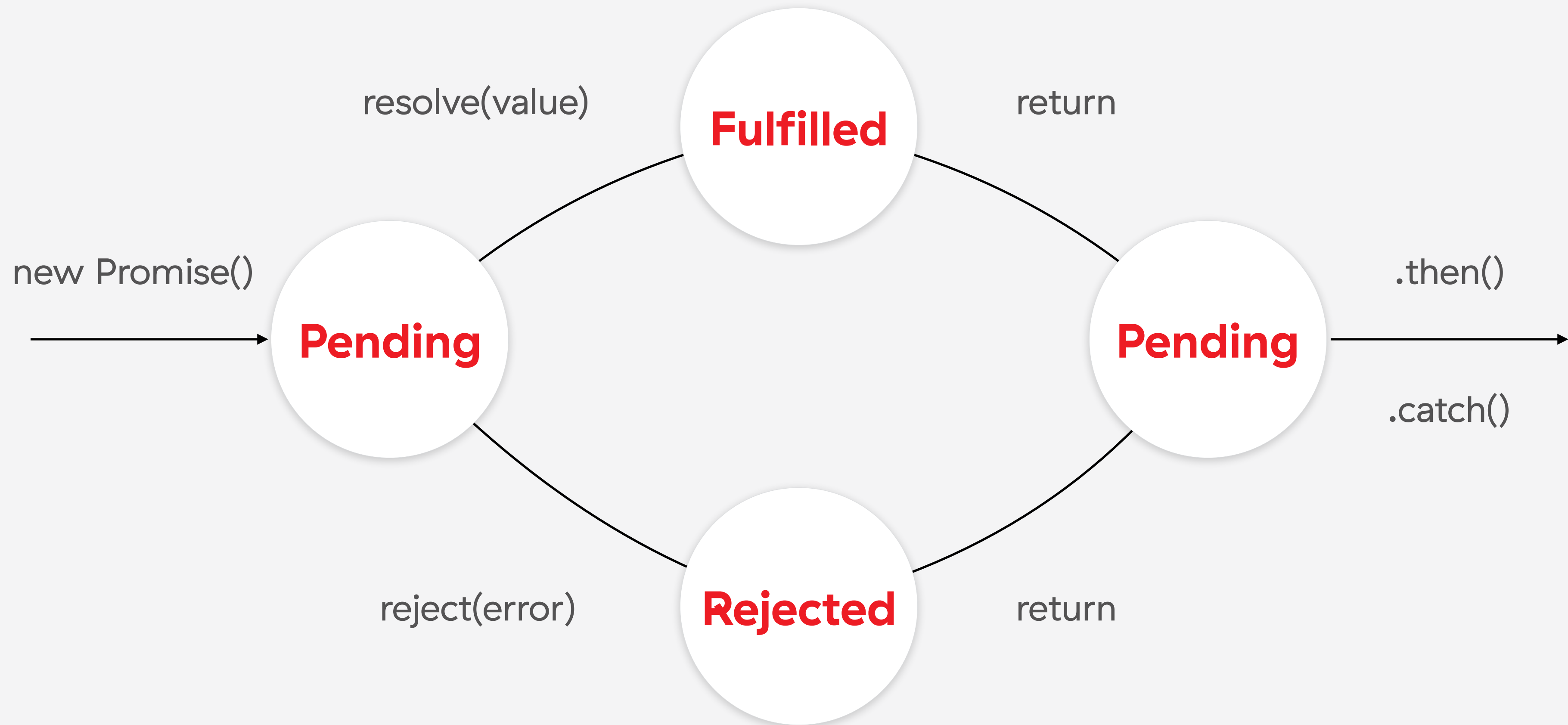
# CHAINING HEAVEN

```
fetchData().then(function(a) {  
    // ...  
}).then(function(b) {  
    // ...  
}).then(function(c) {  
    // ...  
}).then(function(d) {  
    // ...  
}).then(function(e) {  
    // ...  
}).catch(function(err) {  
    // ...  
});
```

# PROMISE METHODS

- `Promise.resolve(value)`
- `Promise.reject(reason)`
- `Promise.all(iterable)`
- ~~`Promise.race(iterable)`~~

# PROMISE STATES



# USAGE

- Available natively in ES6.
- Can be used with libraries such as Q, when, WinJS or RSVP.js, that share a common behaviour called Promises/A+.
- Browser compatibility: 75.69%

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsung Internet
		40	45	5	32	4.1		2.1							
		41	46	5.1	33	4.3		2.2							
		42	47	6	34	5.1		2.3							
		43	48	6.1	35	6.1		3							
6		44	49	7	36	7.1		4							
7		45	50	7.1	37	8		4.1							
8		46	51	8	38	8.4		4.3							
9	12	47	52	9	39	9.2		4.4		12					
10	13	48	53	9.1	40	9.3		4.4.4	7	12.1			10		
11	14	49	54	10	41	10	all	53	10	37	53	49	11	11	4
		50	55	TP	42										
		51	56		43										
		52	57												

- Otherwise use a polyfill

# WHAT'S NEXT: ASYNC/AWAIT

- Part of the ES2017 standard.
- Easier to read and write.
- Available in:
  - Chrome Beta
  - Firefox Nightly
  - Edge – with a flag
  - Safari – in development
- Can be transpiled.

```
async function main() {  
    var quote = await getData();  
    console.log(quote);  
}
```

# CHAINING HEAVEN

```
fetchData().then(function(a) {  
    // ...  
}).then(function(b) {  
    // ...  
}).then(function(c) {  
    // ...  
}).then(function(d) {  
    // ...  
}).then(function(e) {  
    // ...  
}).catch(function(err) {  
    // ...  
});
```

# AWAIT PARADISE

```
async function main() {  
  try {  
    var a = await fetchData();  
    var b = await fetchMoreData(a);  
    var c = await fetchMoreData(b);  
    var d = await fetchMoreData(c);  
    var e = await fetchMoreData(d);  
    // ...  
  } catch(err) {  
    // ...  
  }  
}
```



# CONCLUSION

- Promises allow writing cleaner asynchronous code.
- They have (finally) arrived natively in JavaScript.
- They are well supported by the community.
- You should use them.

## FURTHER READING

- MDN [https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global\\_Objects/Promise](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Promise)
- Promises/A+ <https://promisesaplus.com/>
- JavaScript Promises: an Introduction <https://developers.google.com/web/fundamentals/getting-started/primers/promises>
- We have a problem with promises <https://pouchdb.com/2015/05/18/we-have-a-problem-with-promises.html>



**Thank you!**

Visit [infinum.co](https://infinum.co) or find us on social networks:



[infinum.co](https://infinum.co)



[infinumco](https://twitter.com/infinumco)



[infinumco](https://www.instagram.com/infinumco)



[infinumco](https://www.snapchat.com/add/infinumco)



[infinum](https://www.linkedin.com/company/infinum)