

Warping Time with **Async / Await**

@cassilup, Jun 2017

timjis





Tweet



Hacker News Onion

@HackerNewsOnion



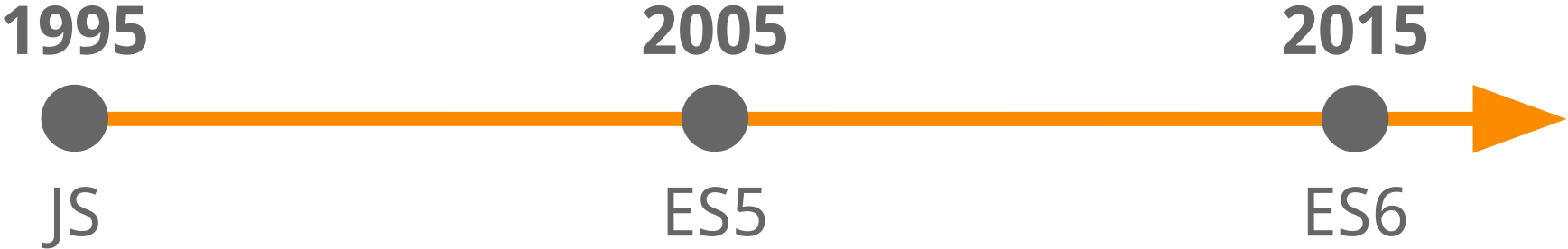
TIL the movie Inception was based on node.js callbacks

12/19/13, 07:42

567 RETWEETS 441 LIKES



JS History (simplified)



JS History (simplified)

1995



JS

2005



ES5

2015



ES6



Page
Load







Ok, I'm grabbing something.

WHEN it appears and **IF** it appears,
I'll show it to you.

JS History (simplified)

1995



JS

Page
Load

2005



ES5

AJAX



2015



ES6



Dictionary

defer¹ | dɪ'fɜː | verb (**defers, deferring, deferred**) [*with obj.*]

put off (an action or event) to a later time; postpone:
they deferred the decision until February.

JS History (simplified)

1995



JS

Page
Load

2005



ES5

AJAX



2015



ES6

Promises



Promises

synchronous

asynchronous

deferred.then(..., ...)

```
1 // ES5
2 function printLatestTweet() {
3     getLatestTweet().then(
4         function (response) {
5             console.log(response);
6         },
7         function (error) {
8             console.error(error);
9         }
10    );
11 };
12
13 // invoke
14 printLatestTweet();
```

```
1 // ES5
2 function printLatestTweet() {
3     getLatestTweet().then(
4         function (response) {
5             console.log(response);
6         },
7         function (error) {
8             console.error(error);
9         }
10    );
11 };
12
13 // invoke
14 printLatestTweet();
```

```
1 // ES5
2 function printLatestTweet() {
3   getLatestTweet().then(
4     function (response) {
5       console.log(response);
6     },
7     function (error) {
8       console.error(error);
9     }
10  );
11 };
12
13 // invoke
14 printLatestTweet();
```

```
1 // ES5
2 function printLatestTweet() {
3     getLatestTweet().then(
4         function (response) {
5             console.log(response);
6         },
7         function (error) {
8             console.error(error);
9         }
10    );
11 };
12
13 // invoke
14 printLatestTweet();
```

syncchronous

asyncchronous

syncchronous

```
1 // ES5
2 function printLatestTweet() {
3     getLatestTweet().then(
4         function (response) {
5             console.log(response);
6         },
7         function (error) {
8             console.error(error);
9         }
10    );
11 };
12
13 // invoke
14 printLatestTweet();
```

```
1 // ES5
2 function printLatestTweet() {
3     getLatestTweet().then(
4         function (response) {
5             console.log(response);
6         },
7         function (error) {
8             console.error(error);
9         }
10    );
11 };
12
13 // invoke
14 printLatestTweet();
```

Promises

synchronous

asynchronous

deferred.then(..., ...)

If only if there was a way to convert the async back into sync.

JS History (there's hope)

1995



JS

Page
Load

2005



ES5

AJAX



2015



ES6

Promises



ES2017

**Async/
Await**

You are here. 👍



ES5

```
1 function printLatestTweet() {
2   getLatestTweet().then(
3     function (response) {
4       console.log(response);
5     }
6   );
7 };
8
9 // invoke
10 printLatestTweet();
```

ES2017

```
1 async function printLatestTweet() {
2   const latestTweet = await getLatestTweet();
3   console.log(latestTweet);
4 };
5
6 // invoke
7 printLatestTweet();
```

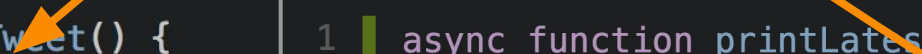
ES5

```
1 function printLatestTweet() {  
2   getLatestTweet().then(  
3     function (response) {  
4       console.log(response);  
5     }  
6   );  
7 };  
8  
9 // invoke  
10 printLatestTweet();
```

ES2017

```
1 async function printLatestTweet() {  
2   const latestTweet = await getLatestTweet();  
3   console.log(latestTweet);  
4 };  
5  
6 // invoke  
7 printLatestTweet();
```

1



ES5

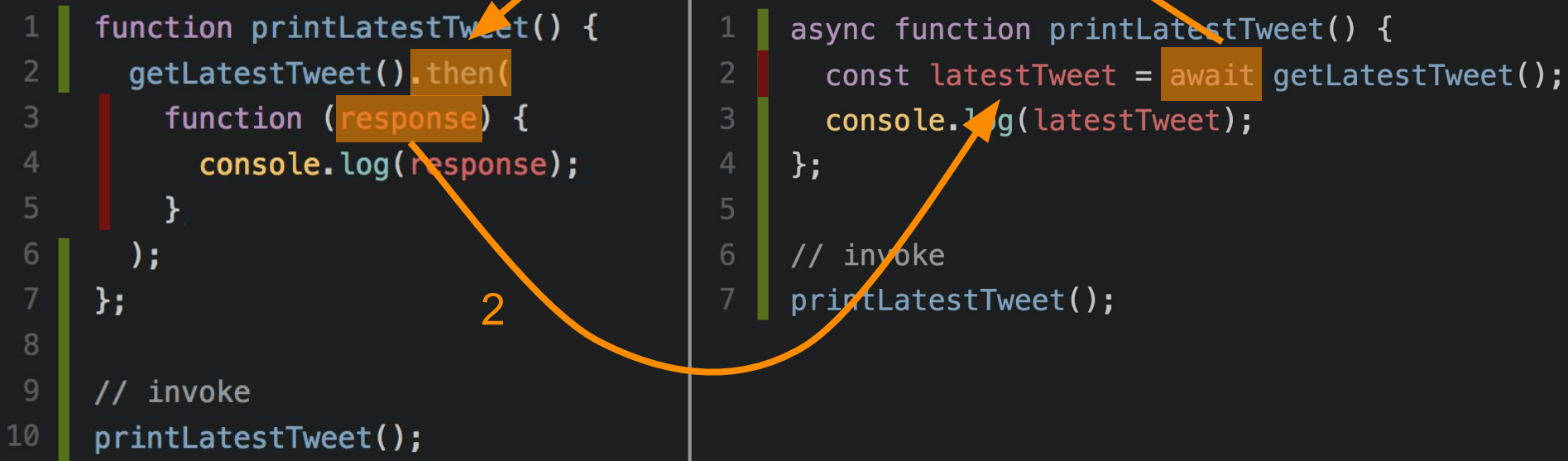
```
1 function printLatestTweet() {  
2   getLatestTweet().then(  
3     function (response) {  
4       console.log(response);  
5     }  
6   );  
7 };  
8  
9 // invoke  
10 printLatestTweet();
```

ES2017

```
1 async function printLatestTweet() {  
2   const latestTweet = await getLatestTweet();  
3   console.log(latestTweet);  
4 };  
5  
6 // invoke  
7 printLatestTweet();
```

2

1



ES5

```
1 function printLatestTweet() {  
2   getLatestTweet().then(  
3     function (response) {  
4       console.log(response);  
5     }  
6   );  
7 };  
8  
9 // invoke  
10 printLatestTweet();
```

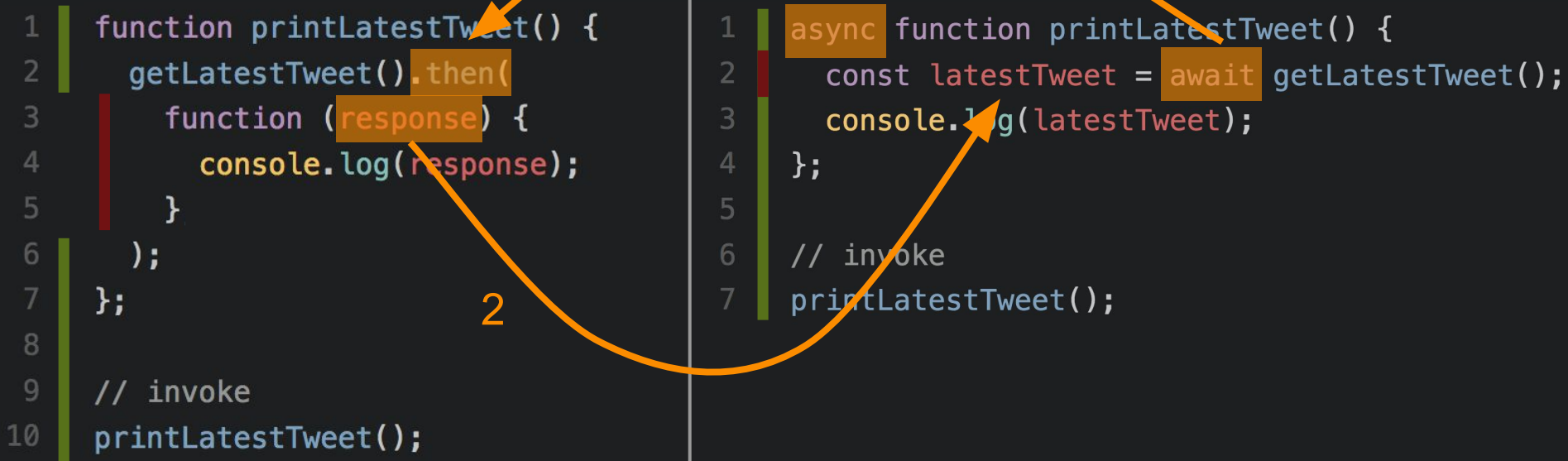
ES2017

```
1 async function printLatestTweet() {  
2   const latestTweet = await getLatestTweet();  
3   console.log(latestTweet);  
4 };  
5  
6 // invoke  
7 printLatestTweet();
```

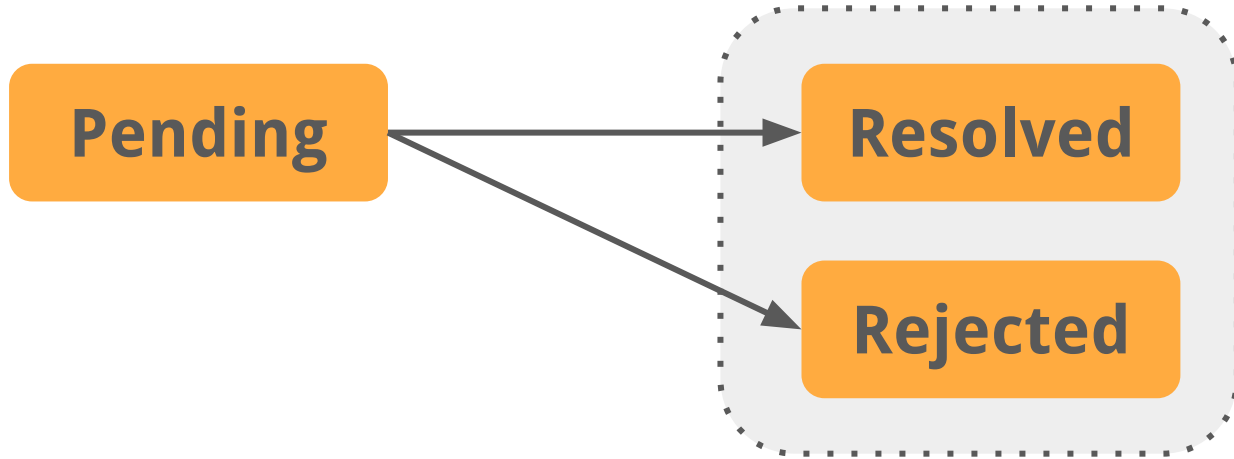
2

3

1



Treating Promise Rejections with `Async/Await`



```
1 // ES2017
2 async function printLatestTweet() {
3   try {
4     const latestTweet = await getLatestTweet();
5     console.log(latestTweet);
6   } catch (error) {
7     console.error(error);
8   }
9 };
10
11 // invoke
12 printLatestTweet();
```

```
1 // ES2017
2 async function printLatestTweet() {
3     try {
4         const latestTweet = await getLatestTweet();
5         console.log(latestTweet);
6     } catch (error) {
7         console.error(error);
8     }
9 };
10
11 // invoke
12 printLatestTweet();
```

As long as it returns a promise,
it can be awaited.



What's in it for me?

tiny.cc/jsheroes-demo

Demo Explained: *Async / Await* codeshift

1. jscodeshift, Facebook tool

<https://github.com/facebook/jscodeshift>

2. Based on ASTs

<https://astexplorer.net>

3. Check out js-codemod

<https://github.com/cpojer/js-codemod>

Async / Await Gotchas

1. **Delayed** execution
 2. **Promise.all()** for making awaited HTTP calls run concurrently
 3. **Non-promises** can also be awaited:
await calls `Promise.resolve` with the value itself.
-

Recap

1. History: Page Load → AJAX → Promises → Async / Await
 2. Syntactic sugar making promises *seem* sync
(promise pitfalls still apply)
 3. Shared state (no more isolated scopes)
 4. async functions return Promises
 5. try-catch is now possible
 6. You can await anything
 7. node.js >= 7.6.0, browsers friendly
-

Thank You.

You guys rock!

@cassilup

Resources

- <https://tc39.github.io/ecmascript-asyncawait/>
- <https://github.com/tc39/ecmascript-asyncawait>
- <https://ponyfoo.com/articles/understanding-javascript-async-await>
- <https://dzone.com/articles/unit-testing-with-jasmine-and-asyncawait>
- <https://applyhead.com/javascript-callback-promise-async-await/>
- <http://thecodebarbarian.com/80-20-guide-to-async-await-in-node.js.html>
- <http://www.jstips.co/en/javascript/improving-your-async-functions-with-webworkers/>
- <http://www.2ality.com/2016/02/async-functions.html>
- <https://zeit.co/blog/async-and-await>
- <https://node.university/blog/498412/es7-es8#async-functions>