Living in harmony

A brief intro to ES6

Maryland.JS Meetup
08/27/2014

- whirlwind tour of ES6 - history, some of the features, resources to get you going
- before I jump into things…

Who am I?

- Based on my gravatar, I’m a super mysterious dude (who wears fingerless gloves)!

- Except I'm not. I'm a husband, father of super-awesome twins

- Huge Orioles fan

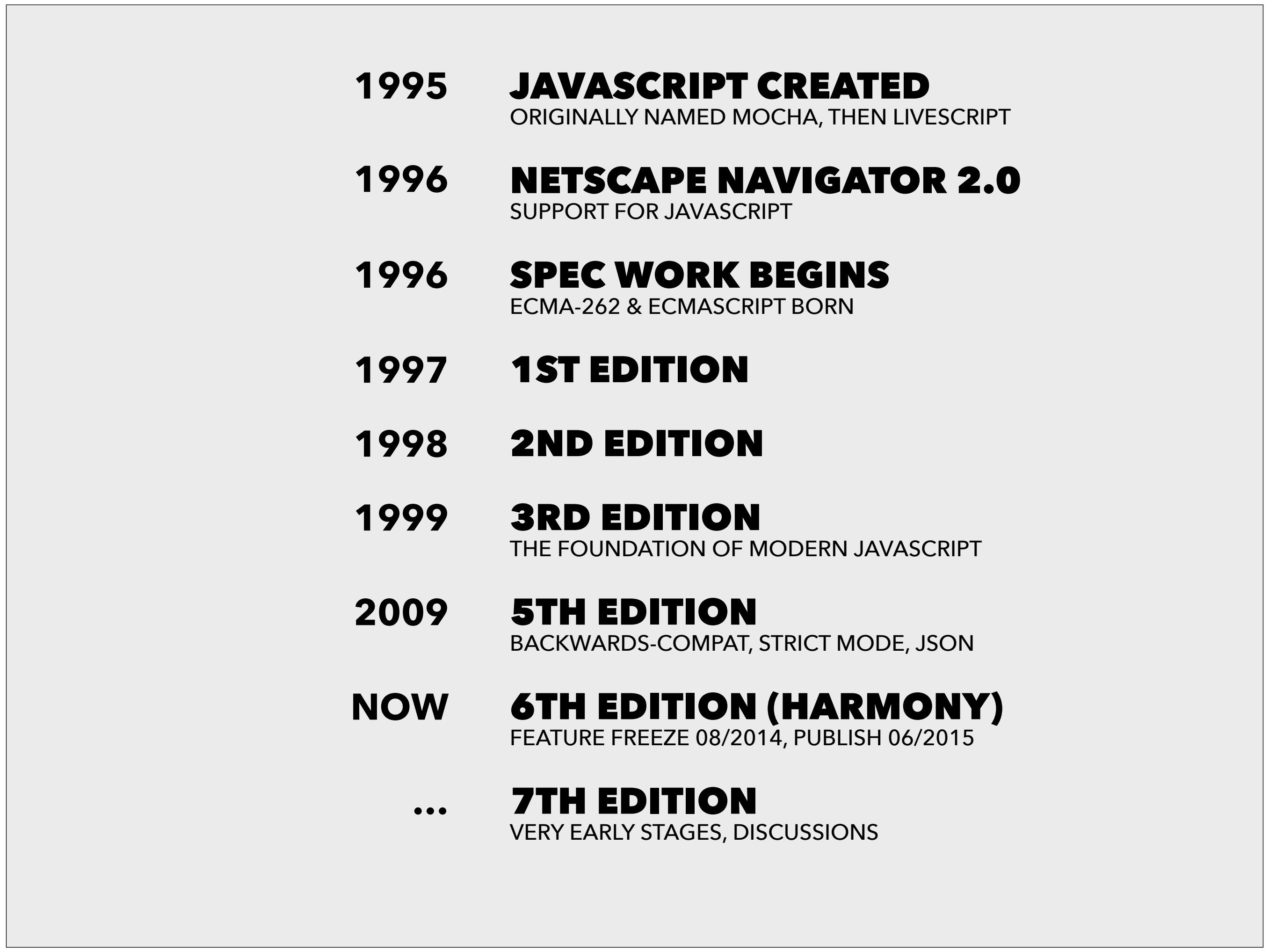
- I work here at Message Systems where we use AngularJS and Node.js daily
- official role: lead software eng

- Unofficial role: defeat these guys at ping pong
- Enough about me…

ECMA-what?

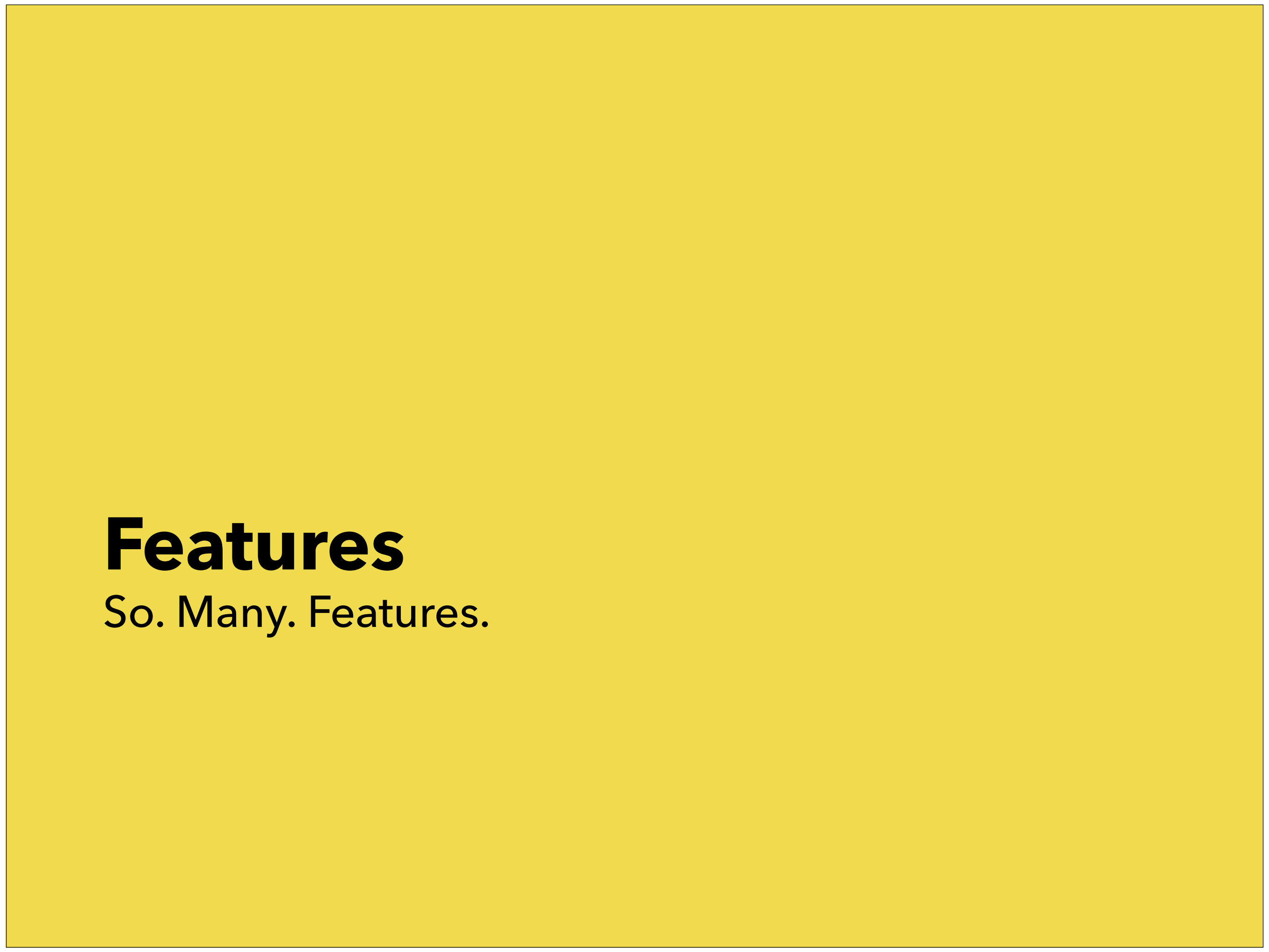
A (very) brief history…

- quick rundown of the history of ECMA script - past, present, future



1995	**JAVASCRIPT CREATED** ORIGINALLY NAMED MOCHA, THEN LIVESCRIPT
1996	**NETSCAPE NAVIGATOR 2.0** SUPPORT FOR JAVASCRIPT
1996	**SPEC WORK BEGINS** ECMA-262 & ECMASCRIPT BORN
1997	**1ST EDITION**
1998	**2ND EDITION**
1999	**3RD EDITION** THE FOUNDATION OF MODERN JAVASCRIPT
2009	**5TH EDITION** BACKWARDS-COMPAT, STRICT MODE, JSON
NOW	**6TH EDITION (HARMONY)** FEATURE FREEZE 08/2014, PUBLISH 06/2015
...	**7TH EDITION** VERY EARLY STAGES, DISCUSSIONS

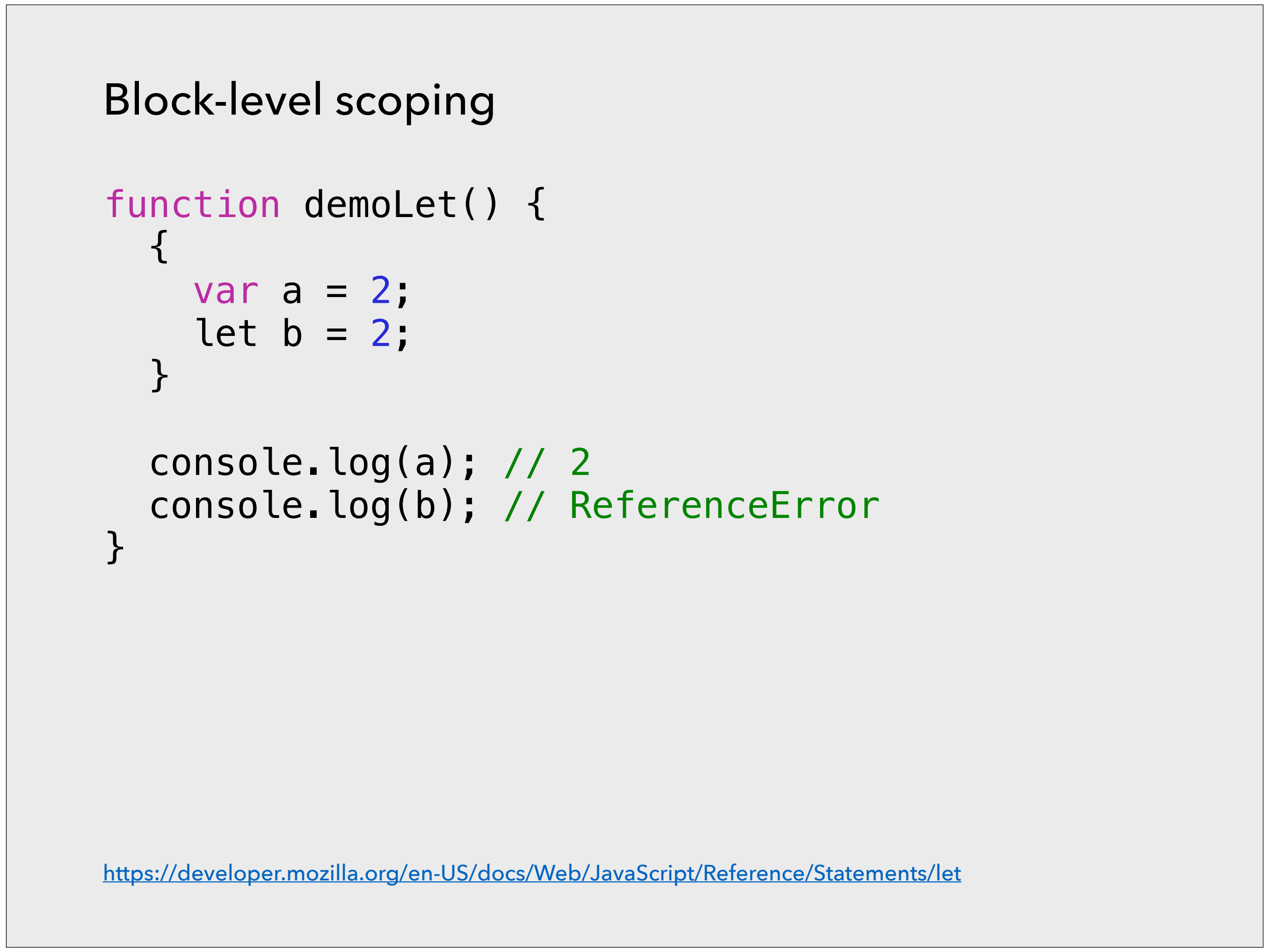
- European Computer Manufacturers Association —> ecma international
- 4th edition started in 2000, abandoned in 2003
- Sources: http://en.wikipedia.org/wiki/Ecma_International, http://ejohn.org/blog/ecmascript-5-strict-mode-json-and-more/, https://twitter.com/awbjs/status/474662357516689410, https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript



Features

So. Many. Features.

- Let’s dive in to the features

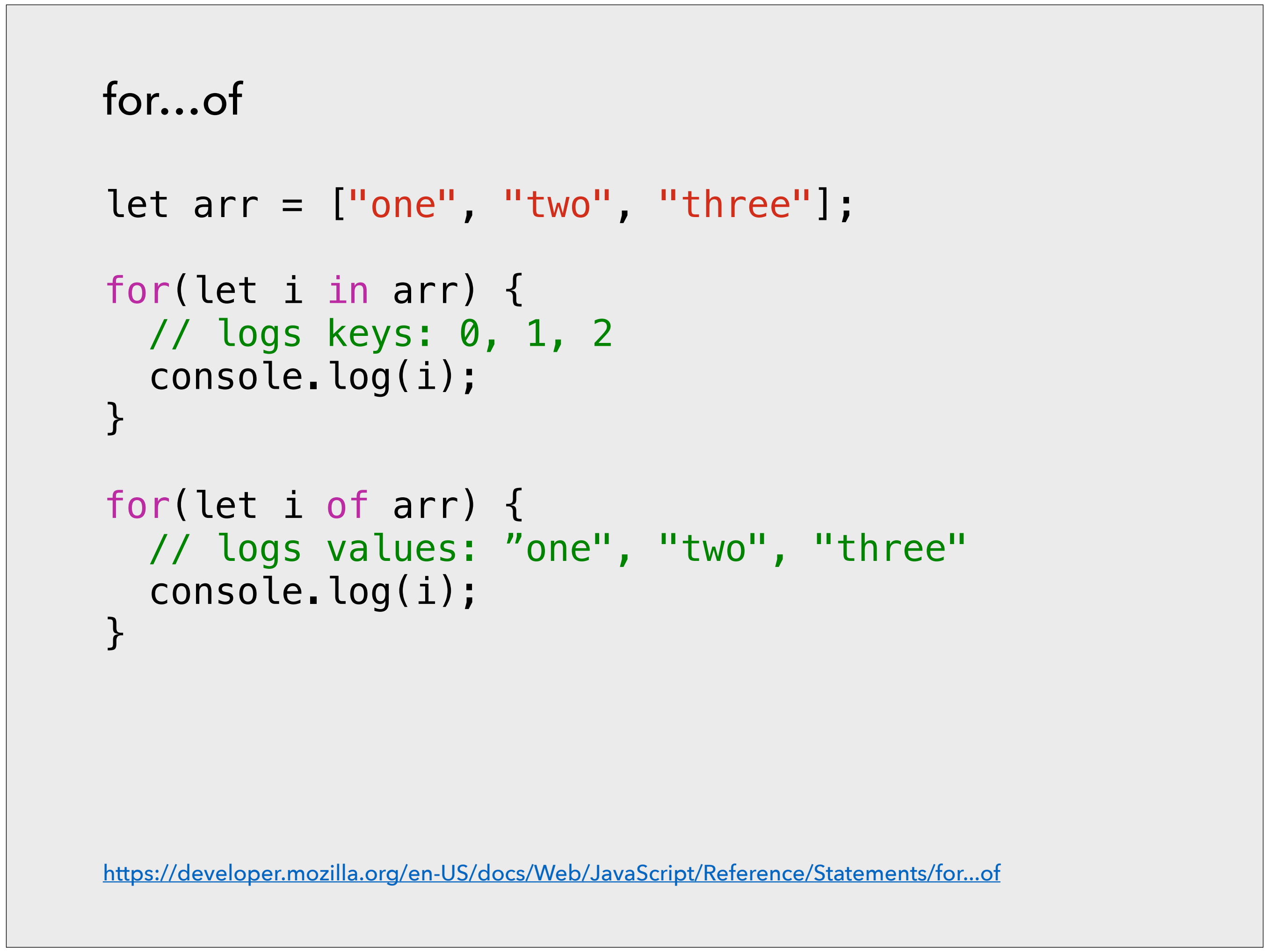


Block-level scoping

```
function demoLet() {
  {
    var a = 2;
    let b = 2;
  }

  console.log(a); // 2
  console.log(b); // ReferenceError
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let

- scoping in JS is.. interesting

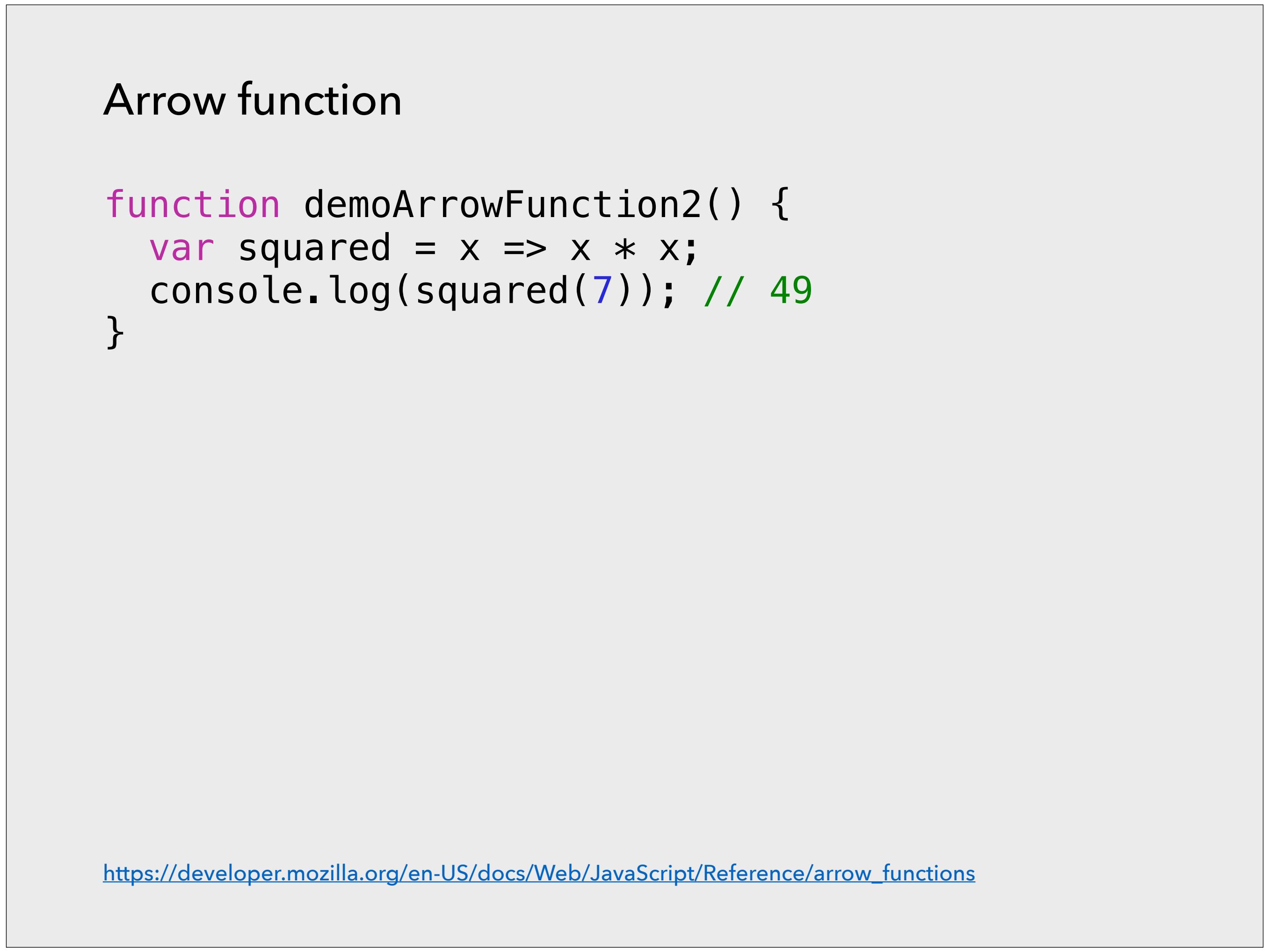


for...of

```
let arr = ["one", "two", "three"];

for(let i in arr) {
  // logs keys: 0, 1, 2
  console.log(i);
}

for(let i of arr) {
  // logs values: ”one", "two", "three"
  console.log(i);
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for...of

- iterate over values
- can also be achieved with forEach

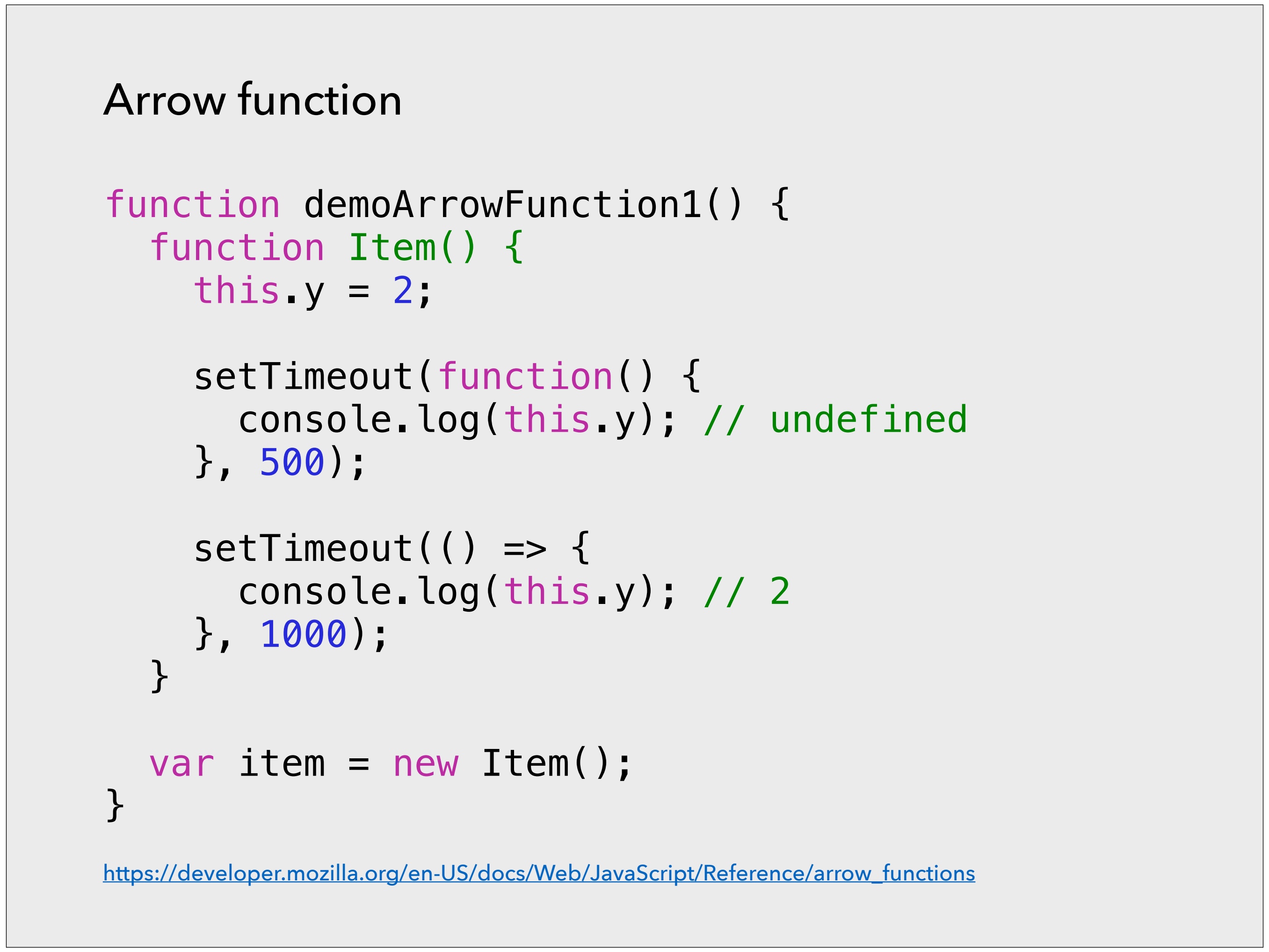


Arrow function

```
function demoArrowFunction2() {
  var squared = x => x * x;
  console.log(squared(7)); // 49
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/arrow_functions

- no need for boilerplate anon functions
- using parameters

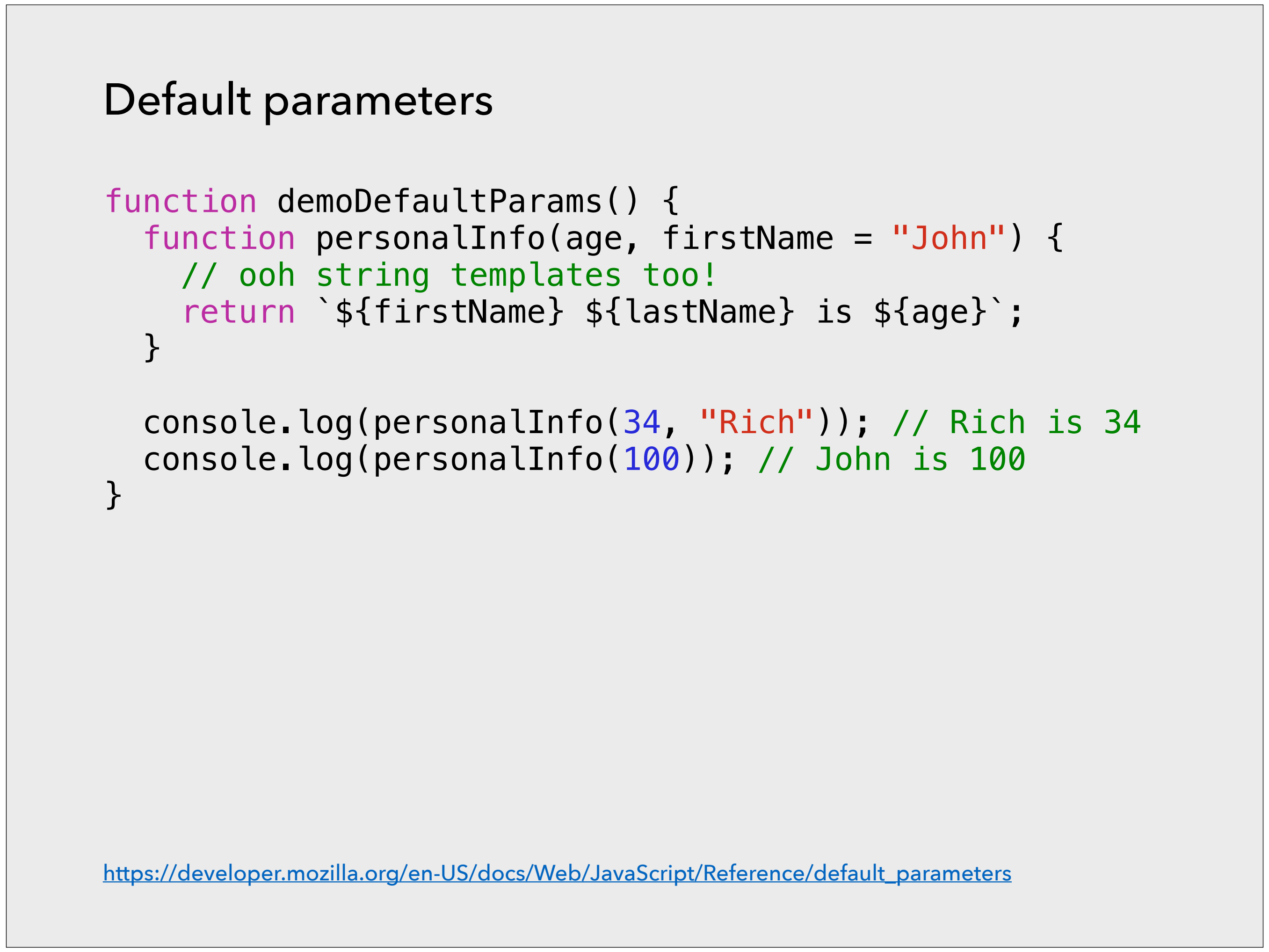


Arrow function

```
function demoArrowFunction1() {
  function Item() {
    this.y = 2;

    setTimeout(function() {
      console.log(this.y); // undefined
    }, 500);

    setTimeout(() => {
      console.log(this.y); // 2
    }, 1000);
  }

  var item = new Item();
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/arrow_functions

- no more self = this (or that = this)
- no params or params, up to you

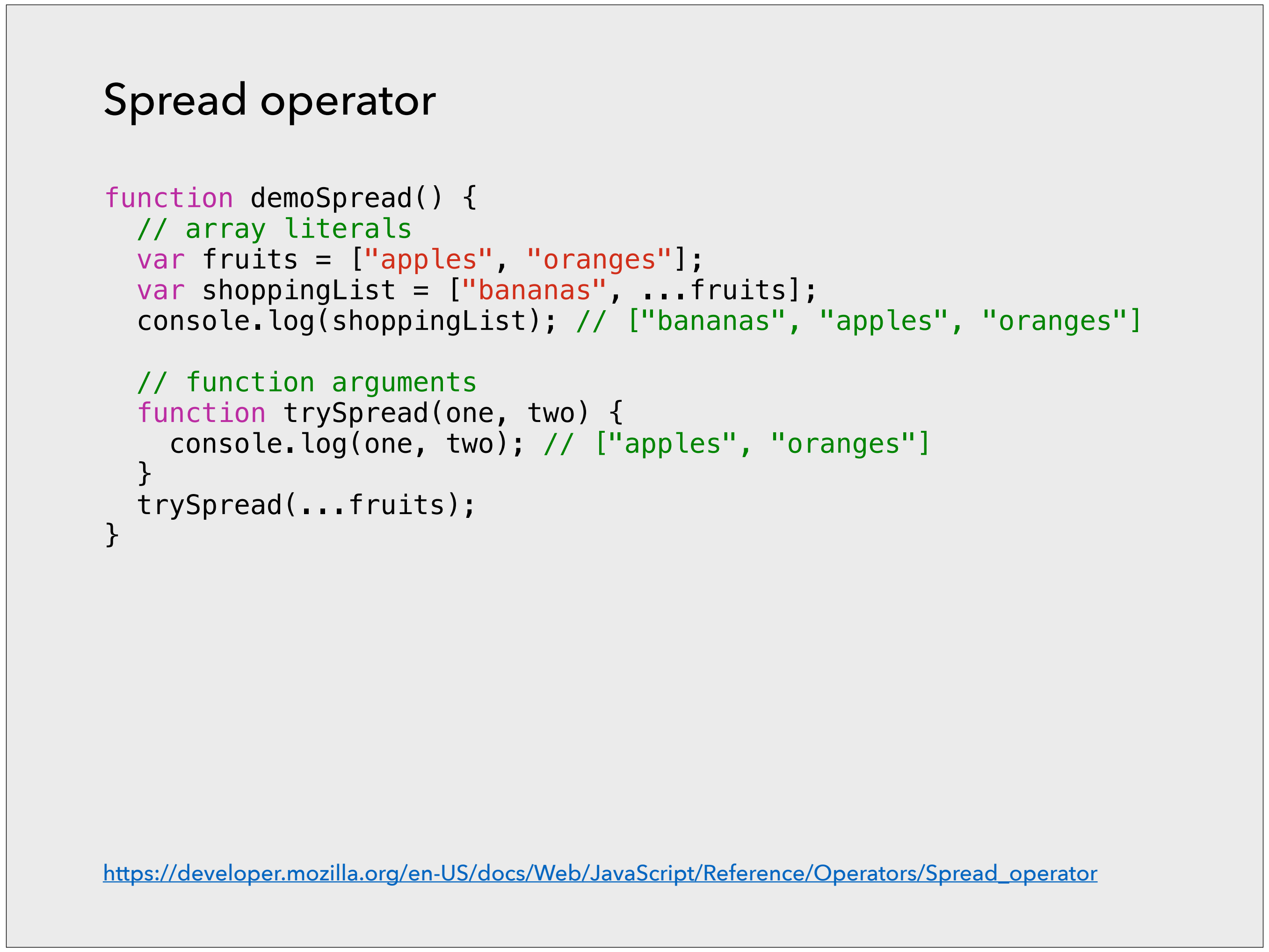


Default parameters

```javascript
function demoDefaultParams() {
  function personalInfo(age, firstName = "John") {
    // ooh string templates too!
    return `${firstName} ${lastName} is ${age}`;
  }

  console.log(personalInfo(34, "Rich")); // Rich is 34
  console.log(personalInfo(100)); // John is 100
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/default_parameters

- coming from other languages like Python - not having this sucks!

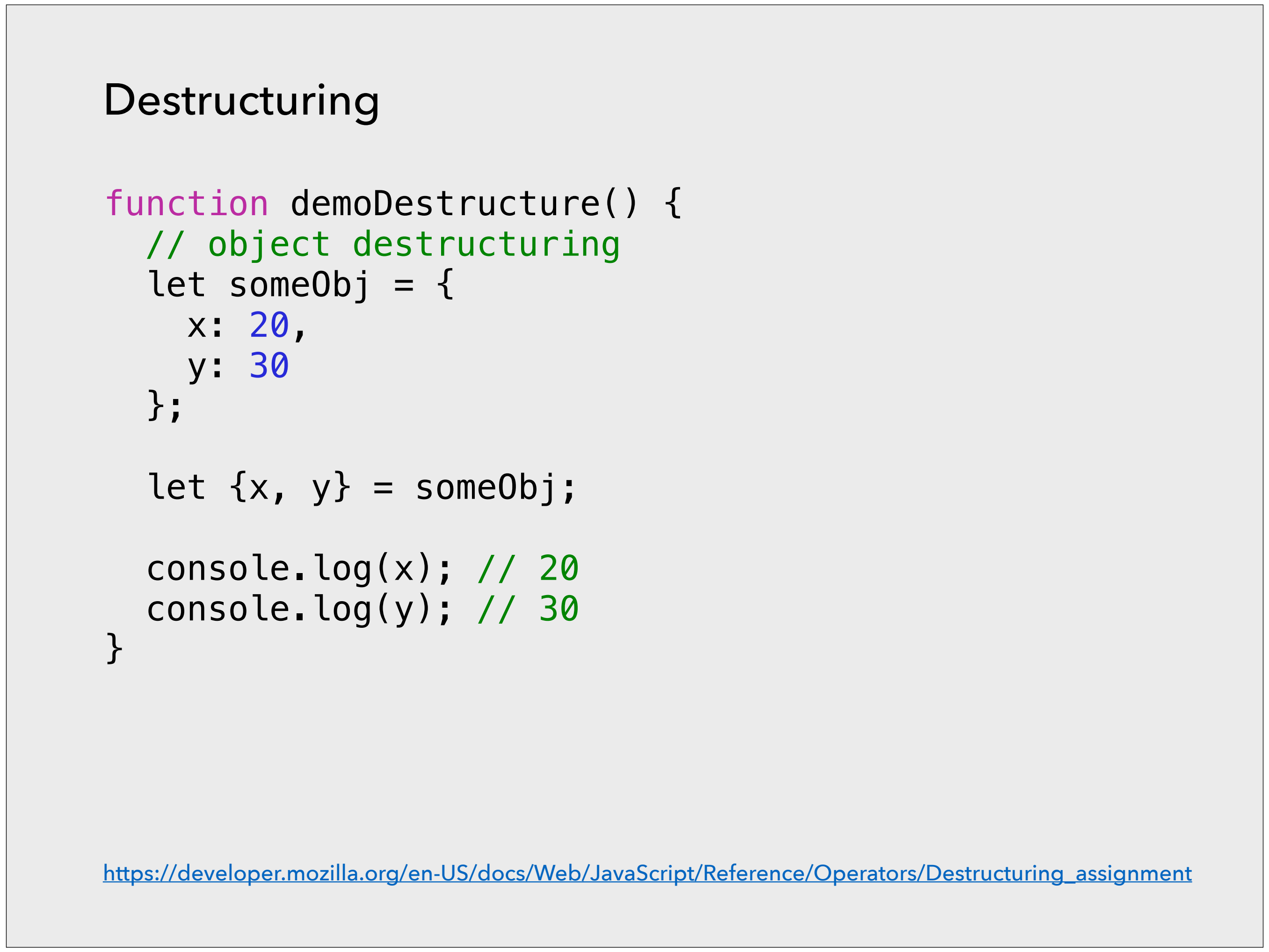


Spread operator

```
function demoSpread() {
  // array literals
  var fruits = ["apples", "oranges"];
  var shoppingList = ["bananas", ...fruits];
  console.log(shoppingList); // ["bananas", "apples", "oranges"]

  // function arguments
  function trySpread(one, two) {
    console.log(one, two); // ["apples", "oranges"]
  }
  trySpread(...fruits);
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_operator

- expansion of multiple arguments

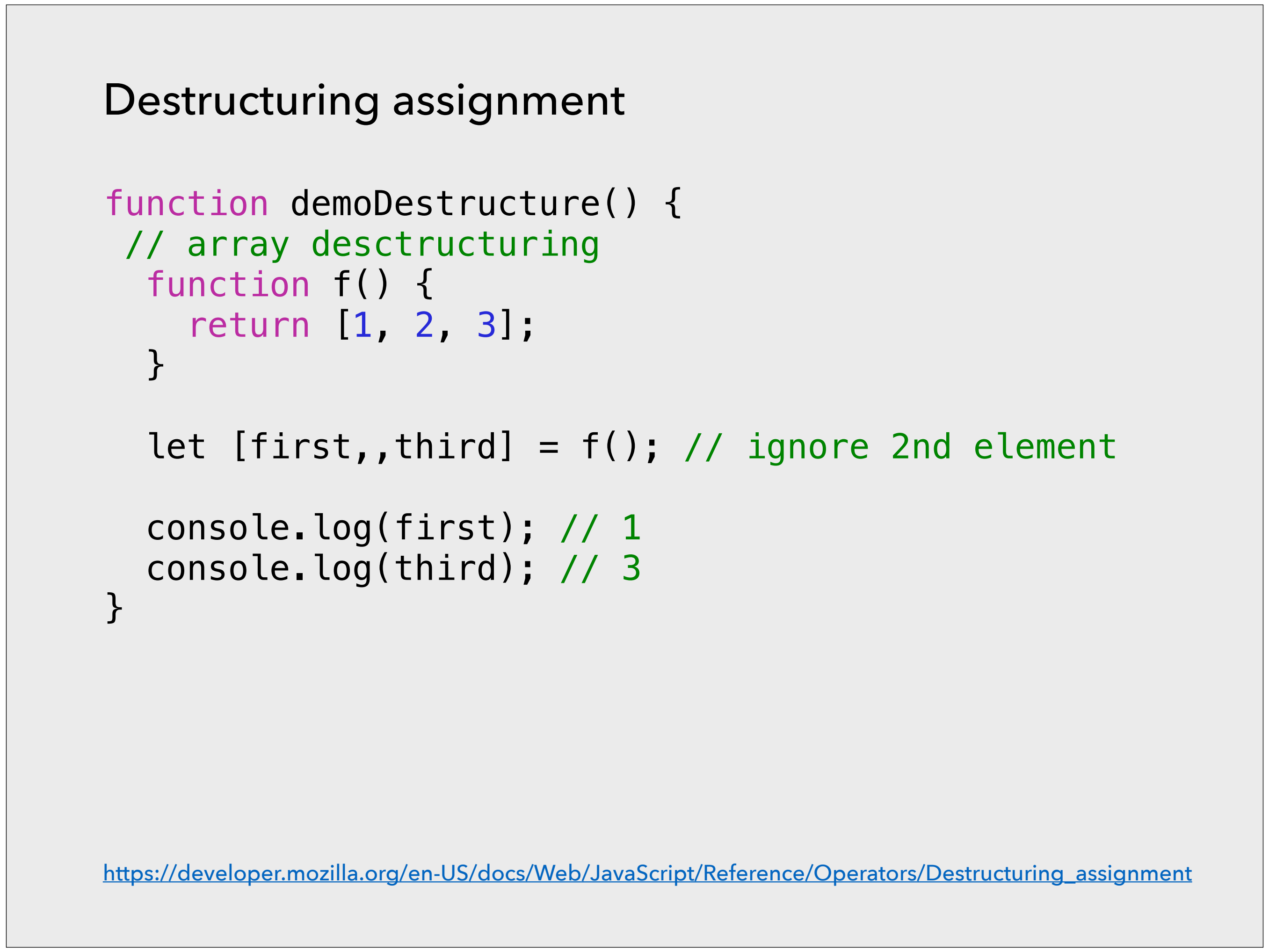


Destructuring

```
function demoDestructure() {
  // object destructuring
  let someObj = {
    x: 20,
    y: 30
  };

  let {x, y} = someObj;

  console.log(x); // 20
  console.log(y); // 30
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment

- xxx

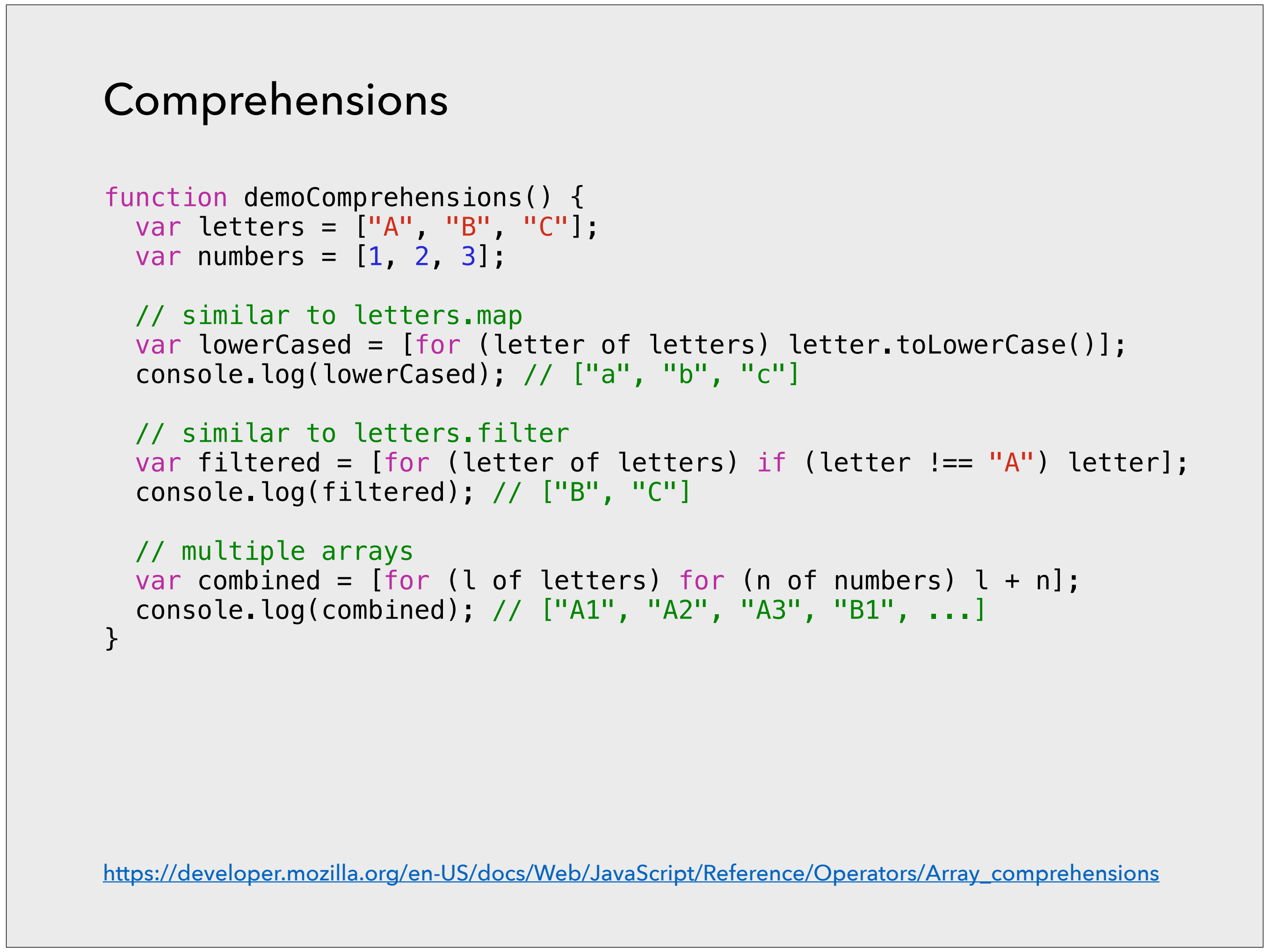


Destructuring assignment

```
function demoDestructure() {
 // array desctructuring
  function f() {
    return [1, 2, 3];
  }

  let [first,,third] = f(); // ignore 2nd element

  console.log(first); // 1
  console.log(third); // 3
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment

- xxx

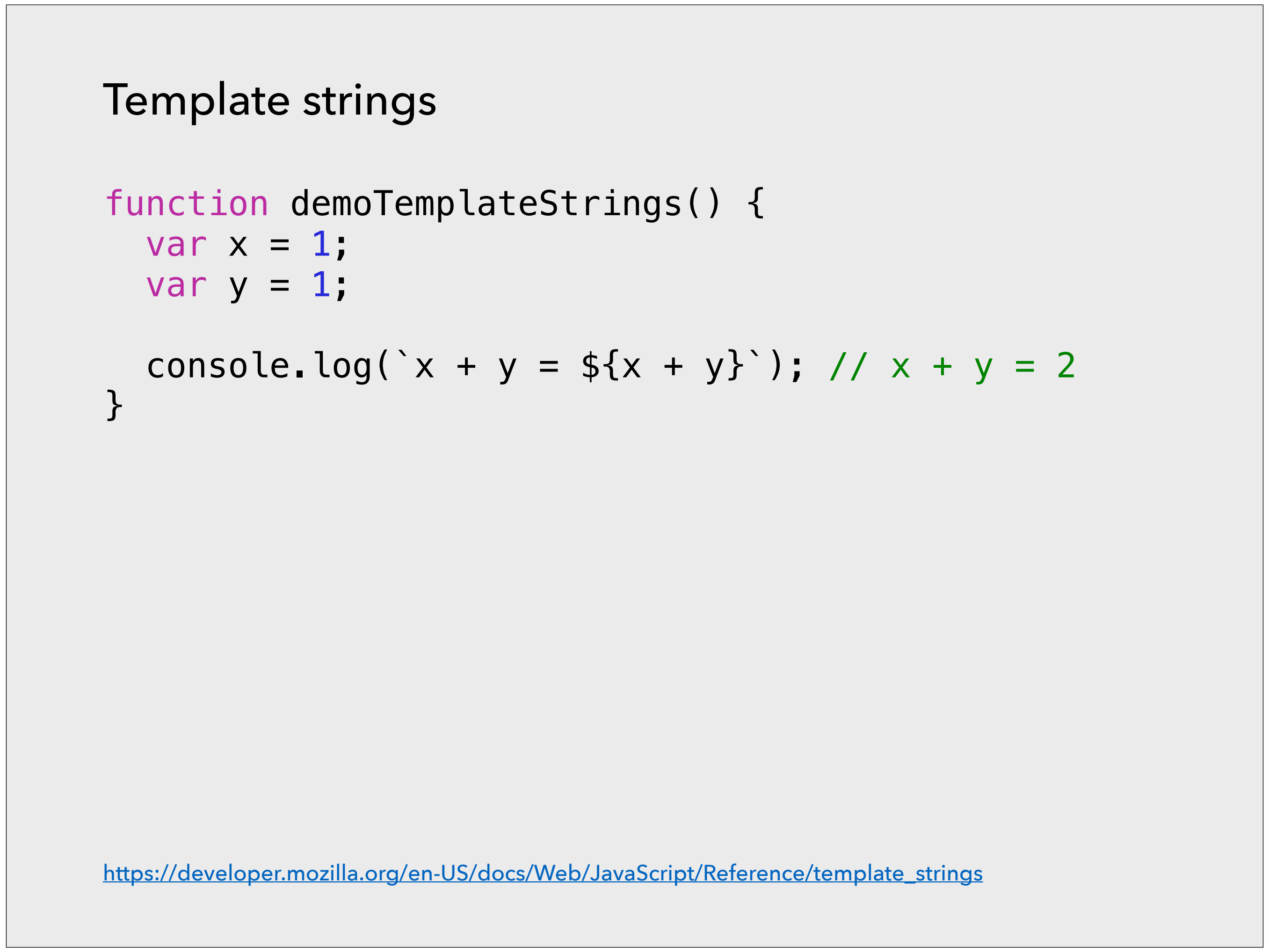


Comprehensions

```
function demoComprehensions() {
  var letters = ["A", "B", "C"];
  var numbers = [1, 2, 3];

  // similar to letters.map
  var lowerCased = [for (letter of letters) letter.toLowerCase()];
  console.log(lowerCased); // ["a", "b", "c"]

  // similar to letters.filter
  var filtered = [for (letter of letters) if (letter !== "A") letter];
  console.log(filtered); // ["B", "C"]

  // multiple arrays
  var combined = [for (l of letters) for (n of numbers) l + n];
  console.log(combined); // ["A1", "A2", "A3", "B1", ...]
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Array_comprehensions

- xxx

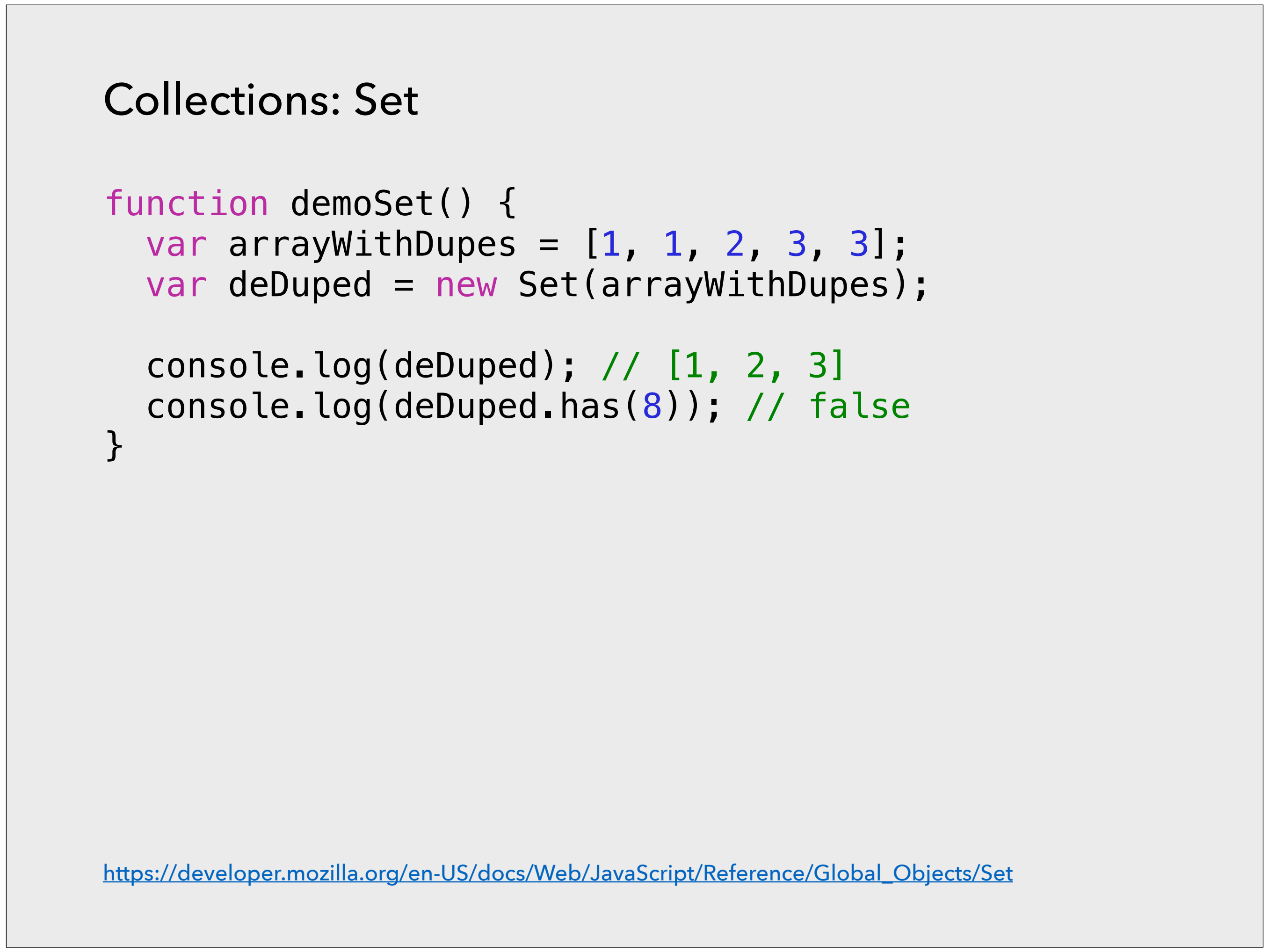


Template strings

```
function demoTemplateStrings() {
  var x = 1;
  var y = 1;

  console.log(`x + y = ${x + y}`); // x + y = 2
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/template_strings

- xxx

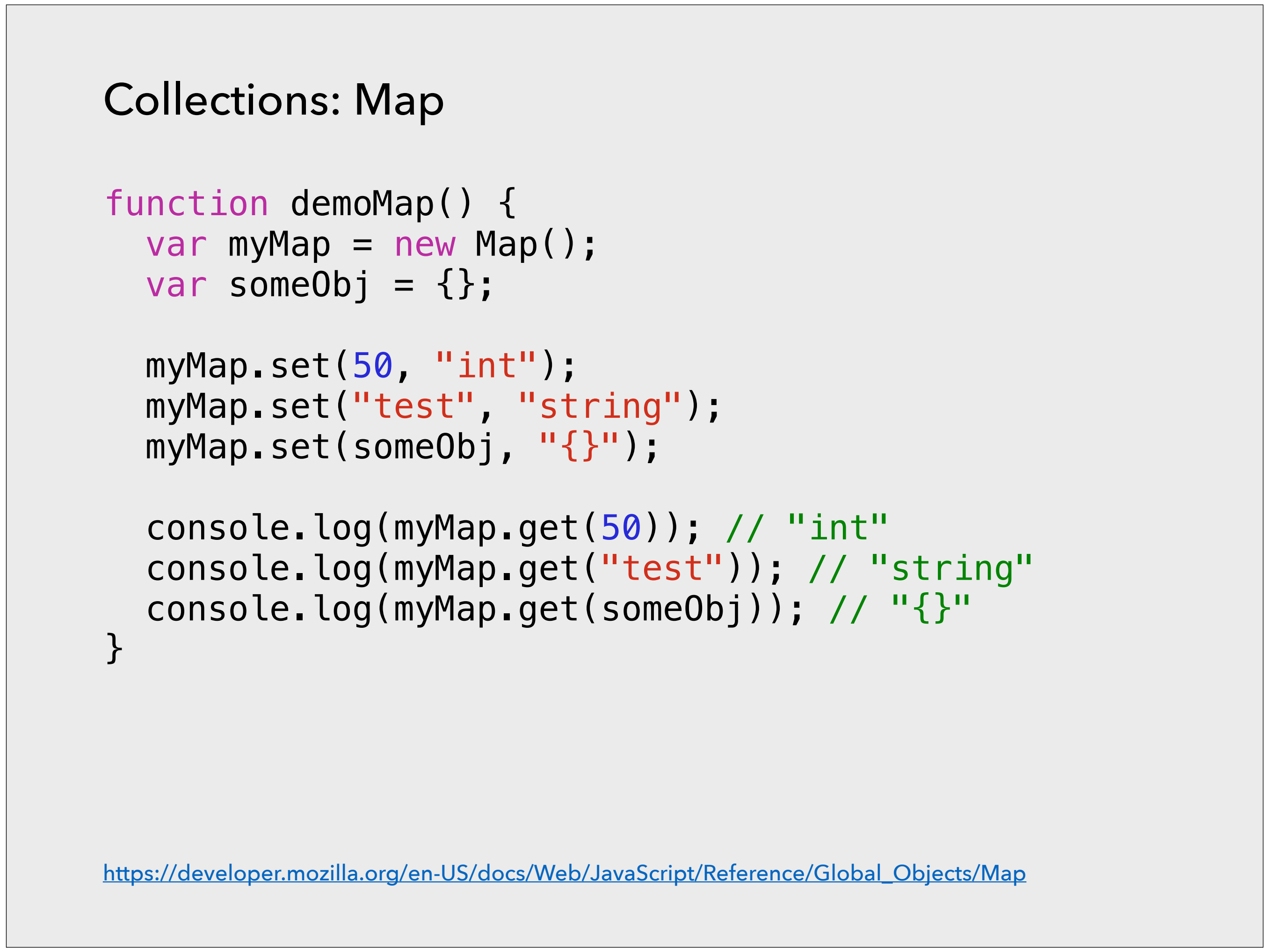


Collections: Set

```
function demoSet() {
  var arrayWithDupes = [1, 1, 2, 3, 3];
  var deDuped = new Set(arrayWithDupes);

  console.log(deDuped); // [1, 2, 3]
  console.log(deDuped.has(8)); // false
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set

- xxx

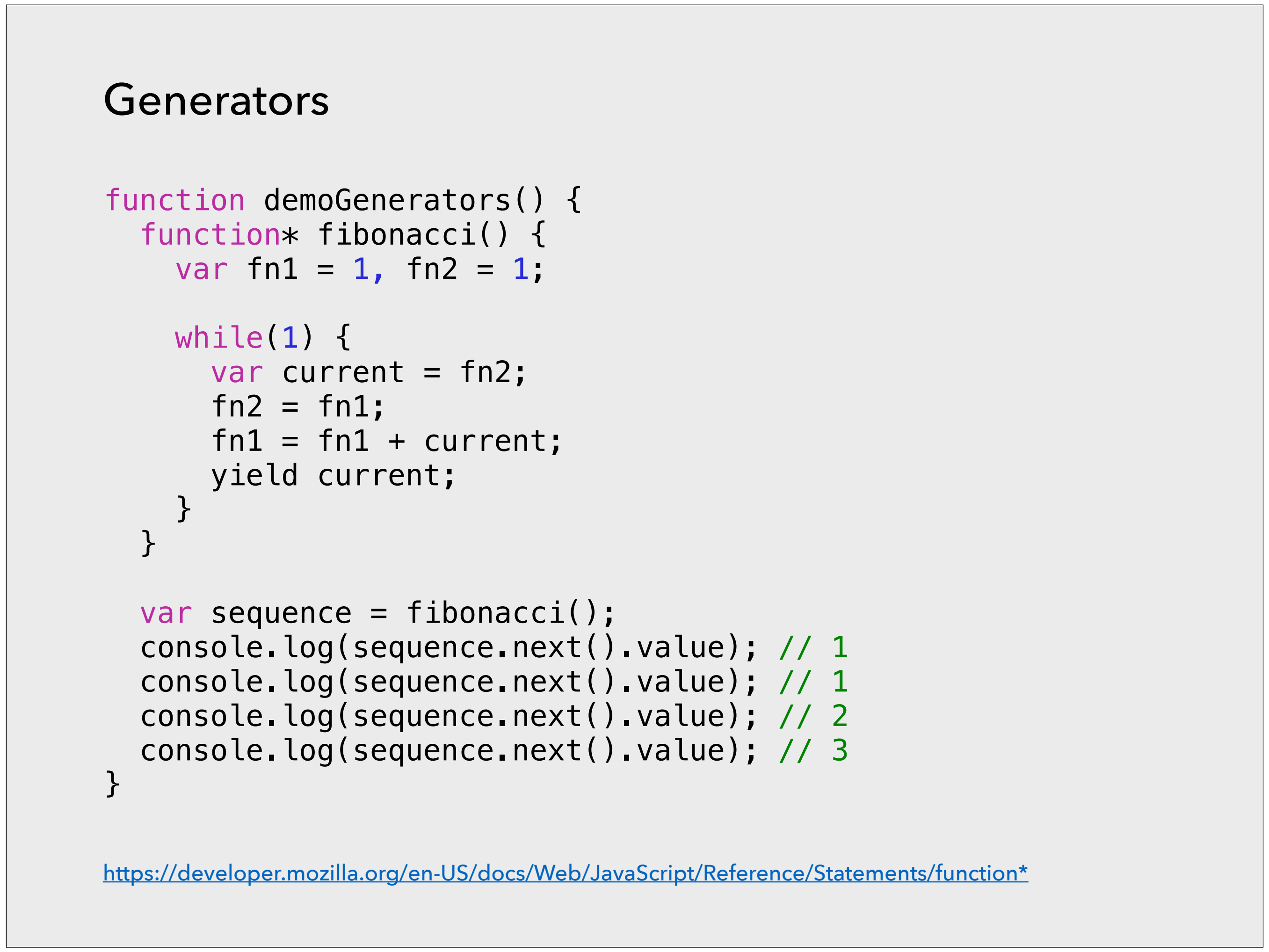


Collections: Map

```
function demoMap() {
  var myMap = new Map();
  var someObj = {};

  myMap.set(50, "int");
  myMap.set("test", "string");
  myMap.set(someObj, "{}");

  console.log(myMap.get(50)); // "int"
  console.log(myMap.get("test")); // "string"
  console.log(myMap.get(someObj)); // "{}"
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Map

- xxx

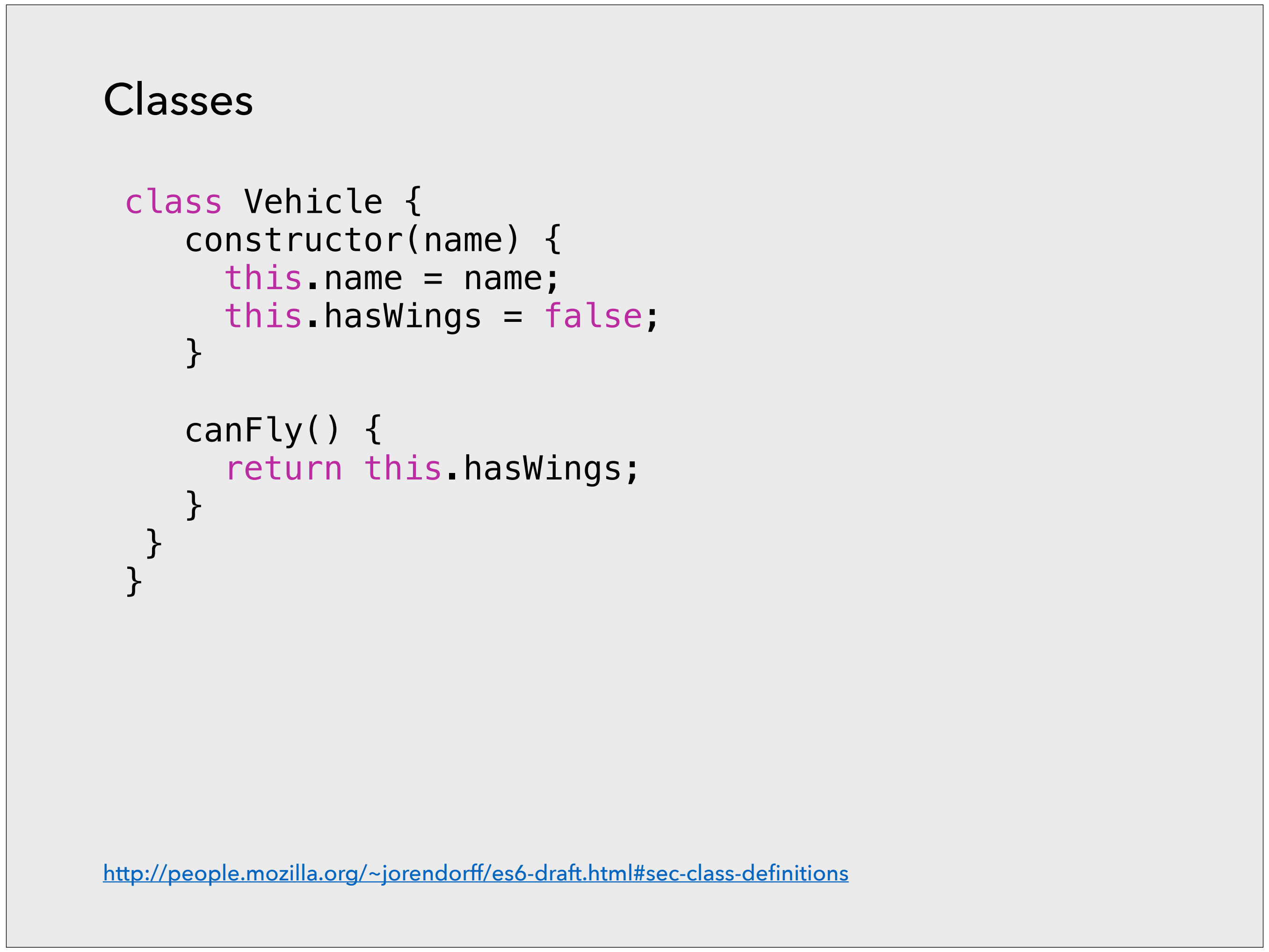


Generators

```
function demoGenerators() {
  function* fibonacci() {
    var fn1 = 1, fn2 = 1;

    while(1) {
      var current = fn2;
      fn2 = fn1;
      fn1 = fn1 + current;
      yield current;
    }
  }

  var sequence = fibonacci();
  console.log(sequence.next().value); // 1
  console.log(sequence.next().value); // 1
  console.log(sequence.next().value); // 2
  console.log(sequence.next().value); // 3
}
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/function*

- xxx



Classes

```
class Vehicle {
   constructor(name) {
     this.name = name;
     this.hasWings = false;
   }

   canFly() {
     return this.hasWings;
   }
 }
}
```

http://people.mozilla.org/~jorendorff/es6-draft.html#sec-class-definitions

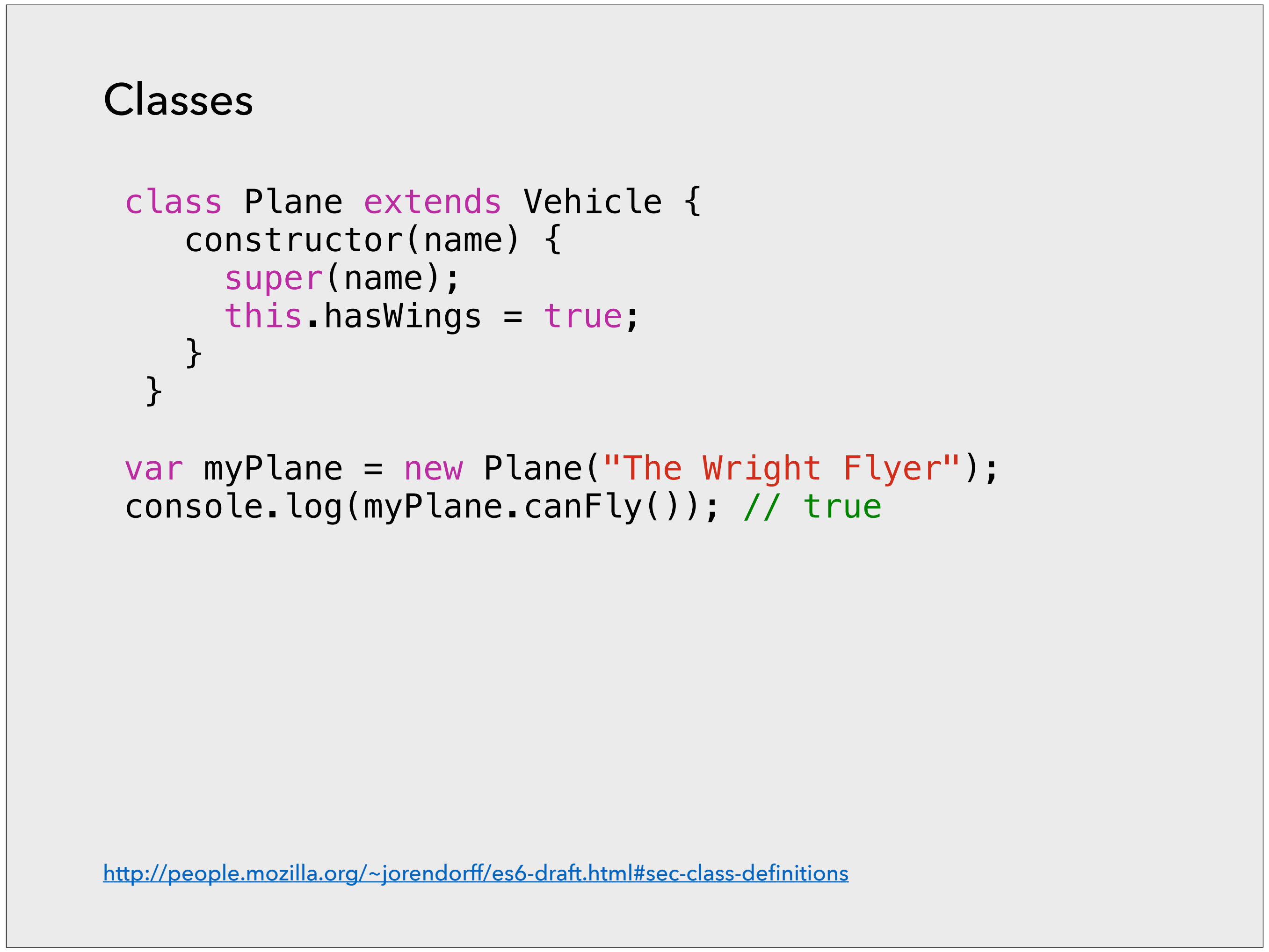
- xxx

Classes

```
class Car extends Vehicle {
   constructor(name, make, model) {
     super(name);
     this.hasWings = false;
     this.make = make;
     this.model = model;
   }
 }

 var myCar = new Car("A-Team Van", "GMC", "Vandura");
 console.log(myCar.canFly()); // false
 console.log(myCar.make); // "GMC"
```

http://people.mozilla.org/~jorendorff/es6-draft.html#sec-class-definitions

- xxx

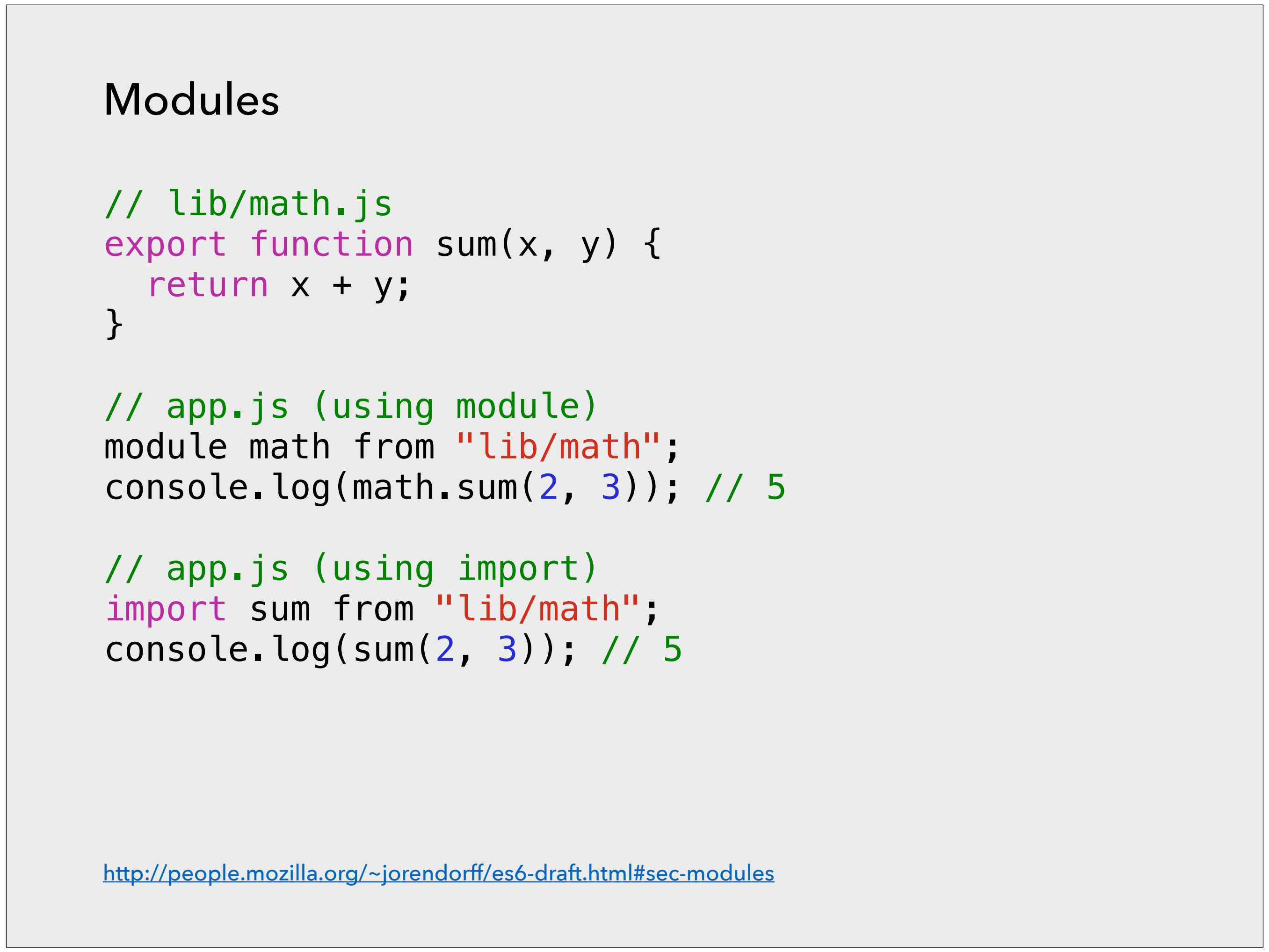


Classes

```
class Plane extends Vehicle {
   constructor(name) {
     super(name);
     this.hasWings = true;
   }
 }

var myPlane = new Plane("The Wright Flyer");
console.log(myPlane.canFly()); // true
```

http://people.mozilla.org/~jorendorff/es6-draft.html#sec-class-definitions

- xxx

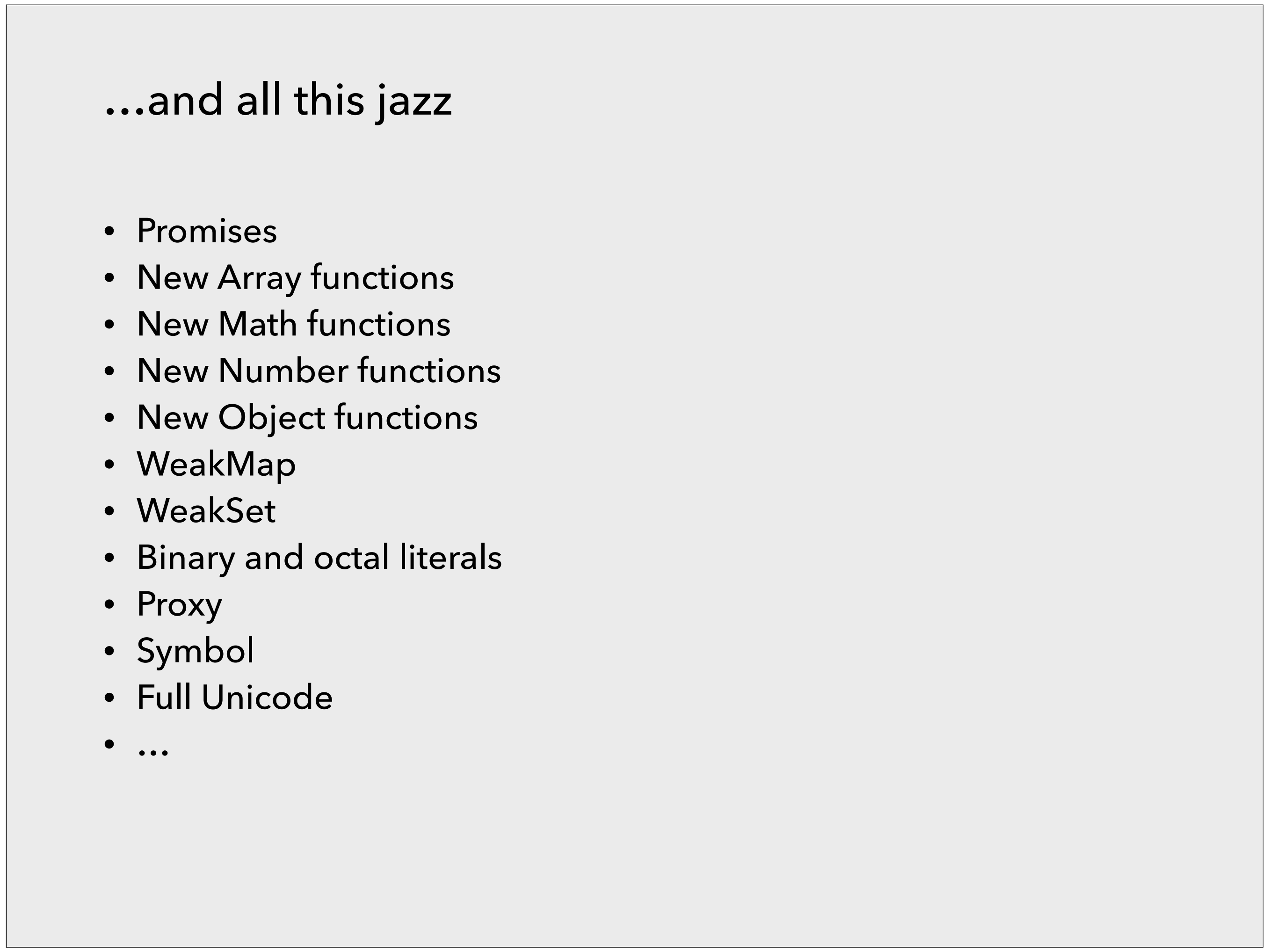


Modules

```
// lib/math.js
export function sum(x, y) {
  return x + y;
}

// app.js (using module)
module math from "lib/math";
console.log(math.sum(2, 3)); // 5

// app.js (using import)
import sum from "lib/math";
console.log(sum(2, 3)); // 5
```

http://people.mozilla.org/~jorendorff/es6-draft.html#sec-modules

- xxx



…and all this jazz

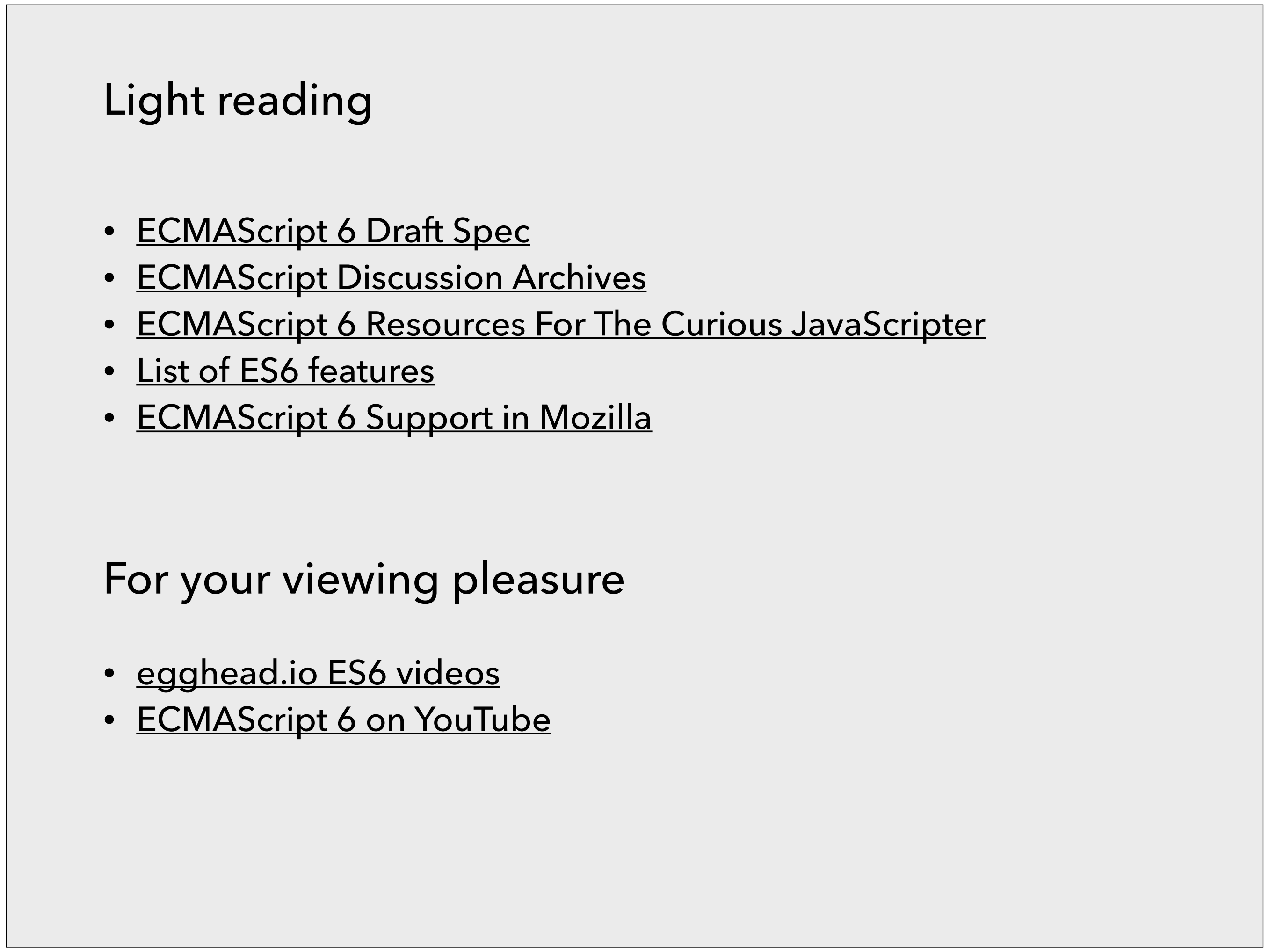
- Promises
- New Array functions
- New Math functions
- New Number functions
- New Object functions
- WeakMap
- WeakSet
- Binary and octal literals
- Proxy
- Symbol
- Full Unicode
- …

- bullet points - other items

Resources

Go on, get!

- xxx



Light reading

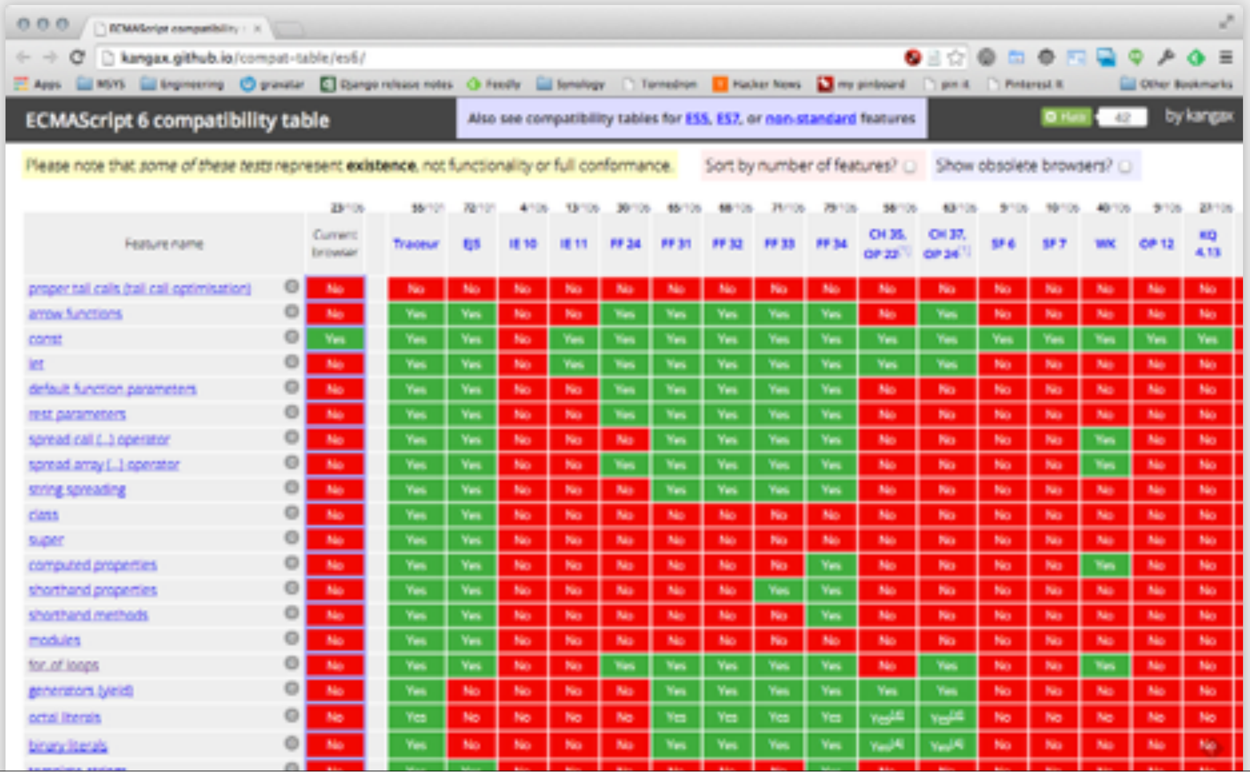
- ECMAScript 6 Draft Spec
- ECMAScript Discussion Archives
- ECMAScript 6 Resources For The Curious JavaScripter
- List of ES6 features
- ECMAScript 6 Support in Mozilla

For your viewing pleasure

- egghead.io ES6 videos
- ECMAScript 6 on YouTube

- tons of resources - search google for es6 resources

ECMAScript 6 compatibility table



kangax.github.io/compat-table/es6/

ECMAScript 6 compatibility table

Also see compatibility tables for ES5, ES7, or non-standard features

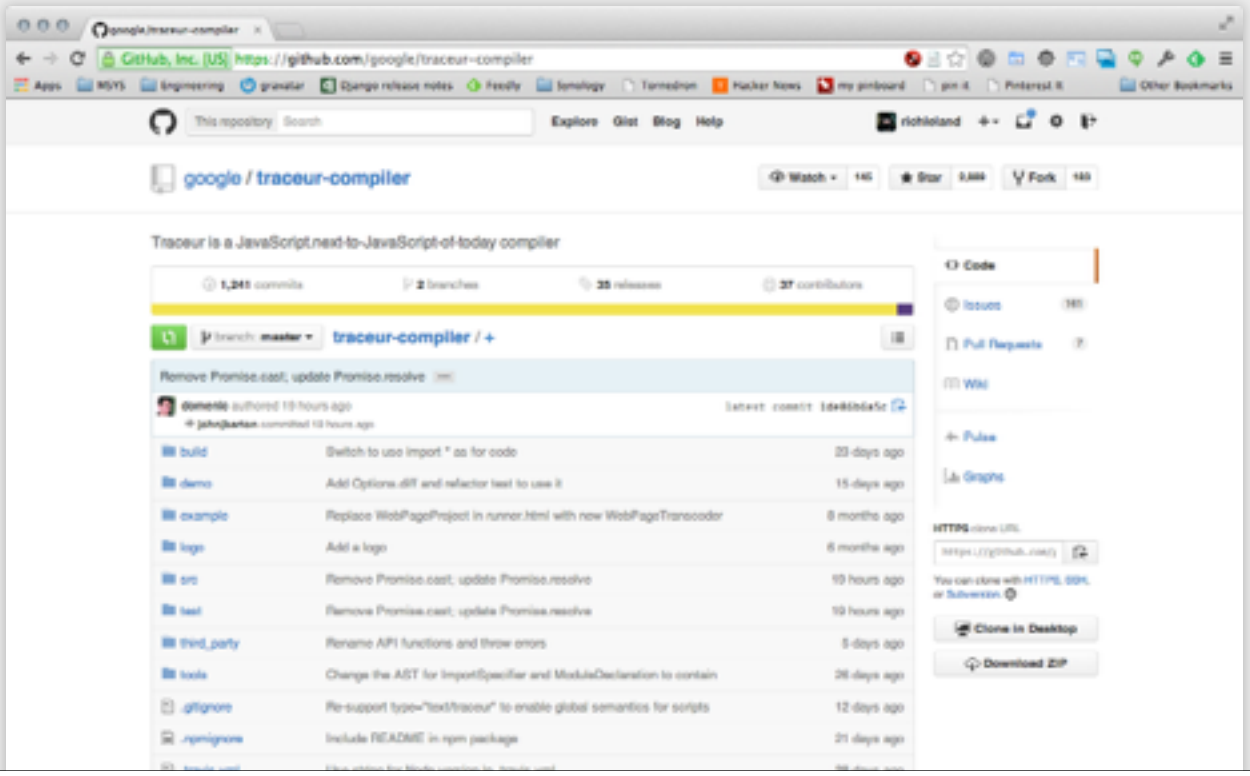
by kangax

Please note that *some of these tests* represent **existence**, not functionality or full conformance.

Sort by number of features? ☐ Show obsolete browsers? ☐

	23/106	55/101	72/101	4/106	13/106	30/106	65/106	68/106	71/106	79/106	58/106	63/106	9/106	10/106	40/106	9/106	27/106
Feature name	Current browser	Traceur	EJS	IE 10	IE 11	FF 24	FF 31	FF 32	FF 33	FF 34	CH 35, OP 22[1]	CH 37, OP 24[1]	SF 6	SF 7	WK	OP 12	KQ 4.13
proper tail calls (tail call optimisation)	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
arrow functions	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No	No
const	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
let	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
default function parameters	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No
rest parameters	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No
spread call (...) operator	No	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	No	No	No	No	Yes	No	No
spread array (...) operator	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	No	No
string spreading	No	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No
class	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No
super	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No
computed properties	No	Yes	Yes	No	No	No	No	No	No	Yes	No	No	No	No	Yes	No	No
shorthand properties	No	Yes	Yes	No	No	No	No	No	Yes	Yes	No	No	No	No	No	No	No
shorthand methods	No	Yes	Yes	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No
modules	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No
for..of loops	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	No	No
generators (yield)	No	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
octal literals	No	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes[4]	Yes[4]	No	No	No	No	No
binary literals	No	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes[4]	Yes[4]	No	No	No	No	No

Traceur compiler



google / traceur-compiler

Traceur is a JavaScript.next-to-JavaScript-of-today compiler

1,241 commits · 2 branches · 35 releases · 37 contributors

branch: master · traceur-compiler / +

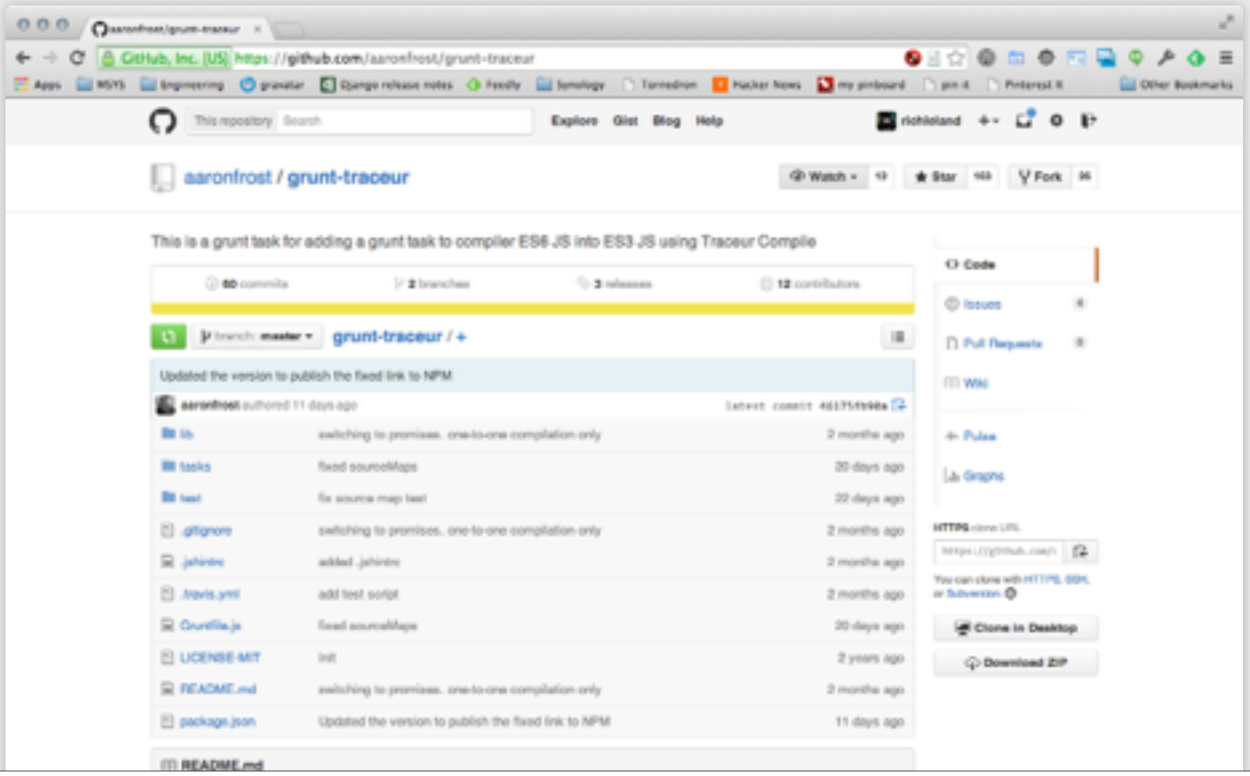
Remove Promise.cast; update Promise.resolve

domenic authored 19 hours ago · latest commit 1de8b0da5c
johnjbarton committed 19 hours ago

build	Switch to use import * as for code	23 days ago
demo	Add Options.diff and refactor test to use it	15 days ago
example	Replace WebPageProject in runner.html with new WebPageTranscoder	8 months ago
logo	Add a logo	6 months ago
src	Remove Promise.cast; update Promise.resolve	19 hours ago
test	Remove Promise.cast; update Promise.resolve	19 hours ago
third_party	Rename API functions and throw errors	5 days ago
tools	Change the AST for ImportSpecifier and ModuleDeclaration to contain	26 days ago
.gitignore	Re-support type="text/traceur" to enable global semantics for scripts	12 days ago
.npmignore	Include README in npm package	21 days ago

You can clone with HTTPS, SSH, or Subversion.

grunt-traceur-latest


https://github.com/aaronfrost/grunt-traceur

aaronfrost / grunt-traceur

This is a grunt task for adding a grunt task to compiler ES6 JS into ES3 JS using Traceur Compile

60 commits · 2 branches · 3 releases · 12 contributors

branch: master · grunt-traceur / +

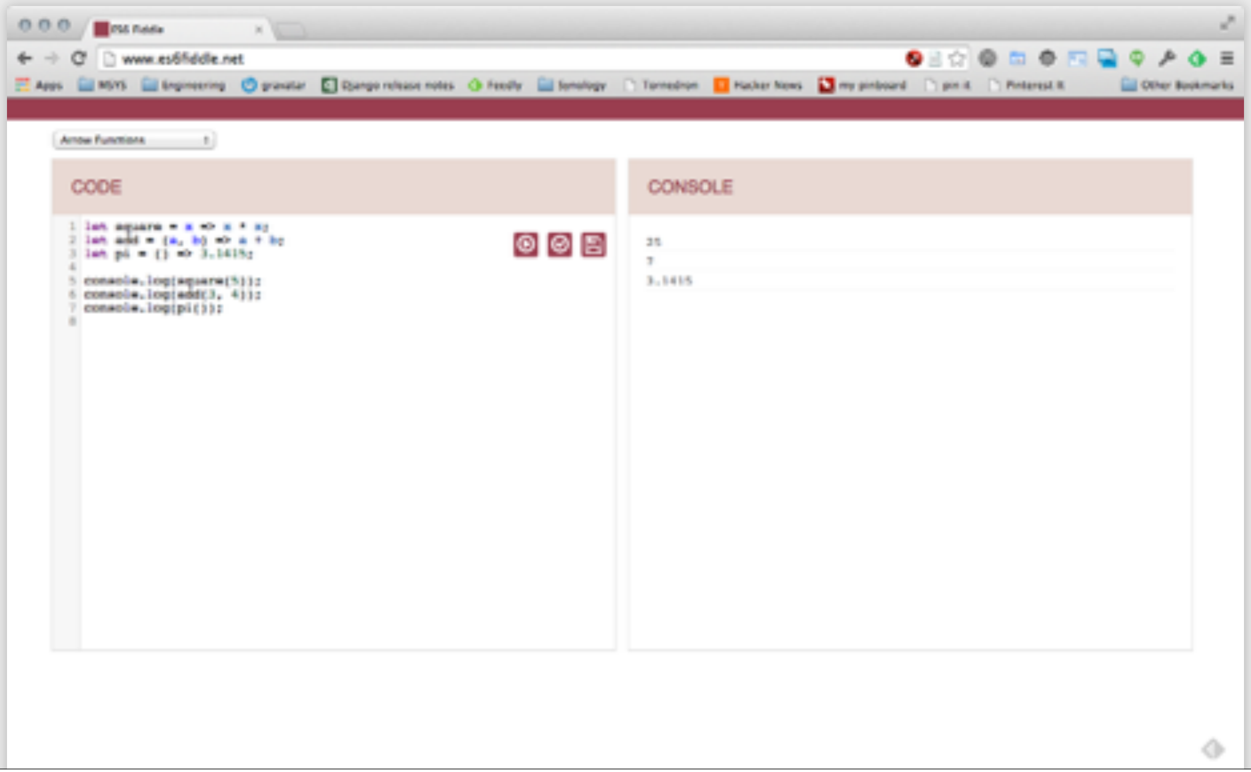
Updated the version to publish the fixed link to NPM

aaronfrost authored 11 days ago · latest commit 461f5fb98a

lib	switching to promises. one-to-one compilation only	2 months ago
tasks	fixed sourceMaps	20 days ago
test	fix source map test	20 days ago
.gitignore	switching to promises. one-to-one compilation only	2 months ago
.jshintrc	added .jshintrc	2 months ago
.travis.yml	add test script	2 months ago
Gruntfile.js	fixed sourceMaps	20 days ago
LICENSE-MIT	init	2 years ago
README.md	switching to promises. one-to-one compilation only	2 months ago
package.json	Updated the version to publish the fixed link to NPM	11 days ago

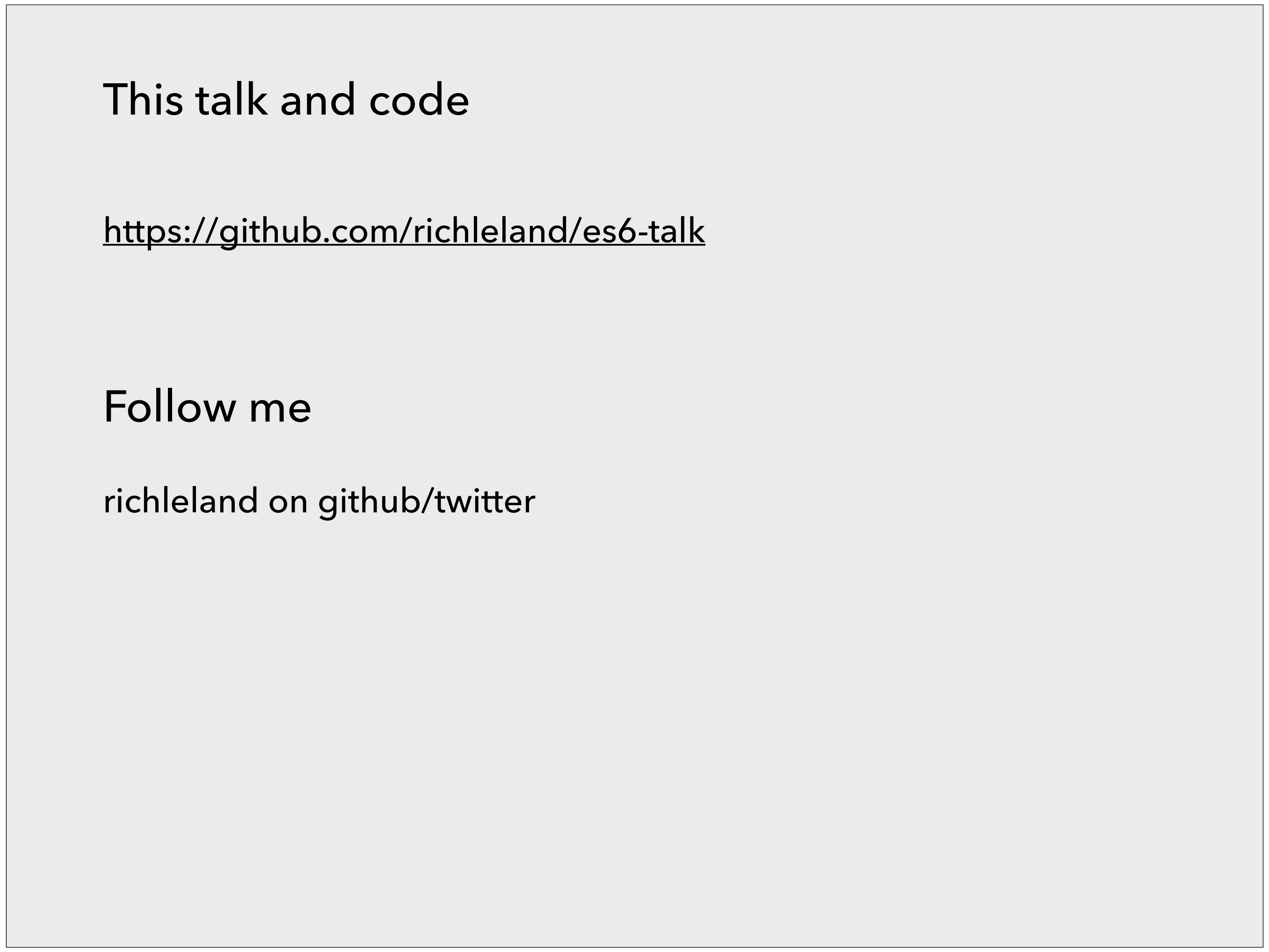
README.md

http://www.es6fiddle.net/



```
let square = x => x * x;
let add = (a, b) => a + b;
let pi = () => 3.1415;

console.log(square(5));
console.log(add(3, 4));
console.log(pi());
```

```
25
7
3.1415
```



This talk and code

https://github.com/richleland/es6-talk

Follow me

richleland on github/twitter

- xxx

- xxx