EcmaScript 6

Jeff McRiffey
@jmcriffey

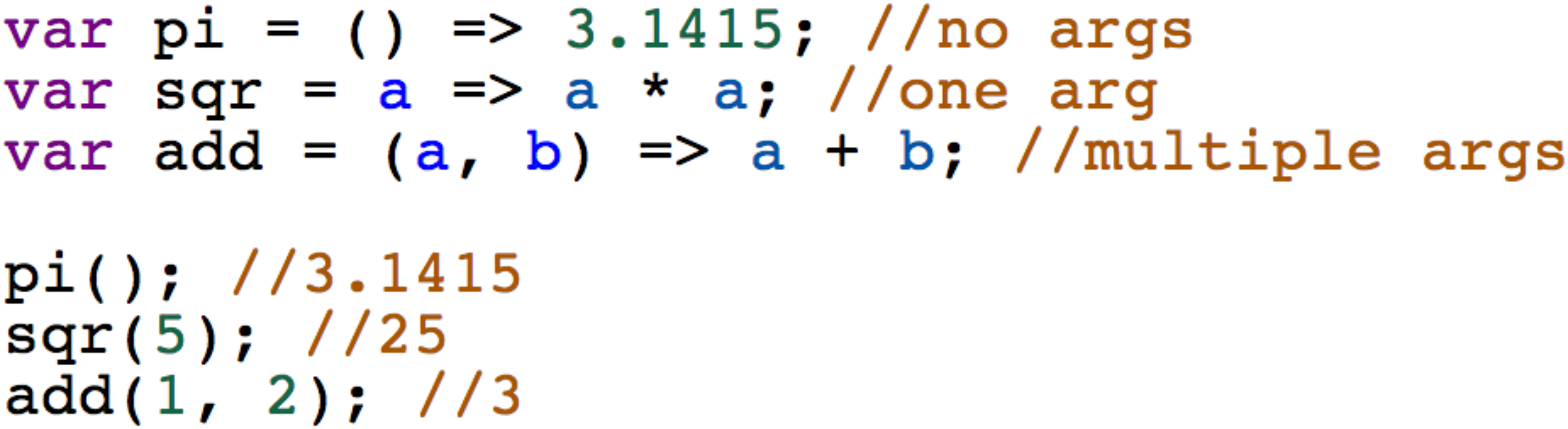
Outline

- Intro
- Features
- Tools
- Future

Introduction

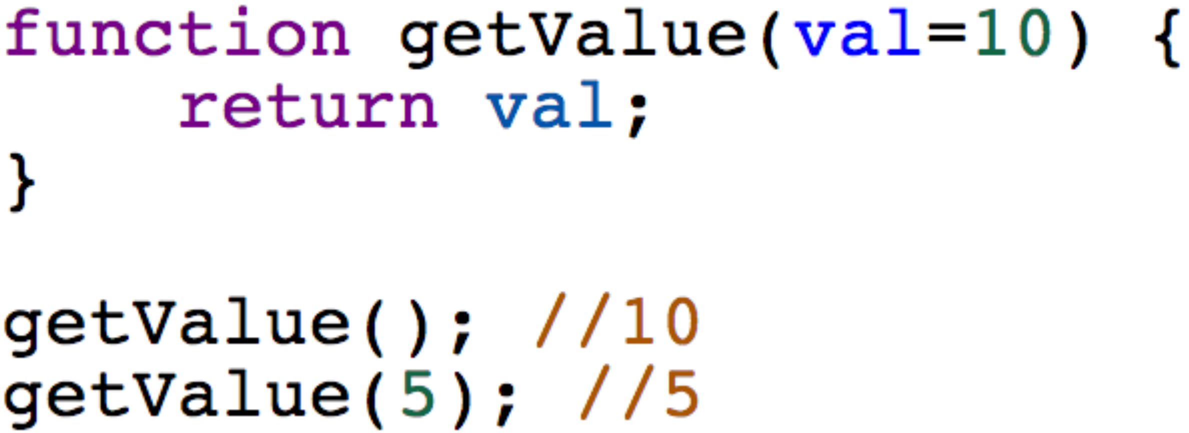
- ECMAScript
    - Spec for JavaScript
    - ECMAScript 5
        - Published late 2009
    - ECMAScript 6 (harmony)
        - Expected to publish in June 2015
        - Feature set is frozen
        - Browsers already implementing

Arrow Functions



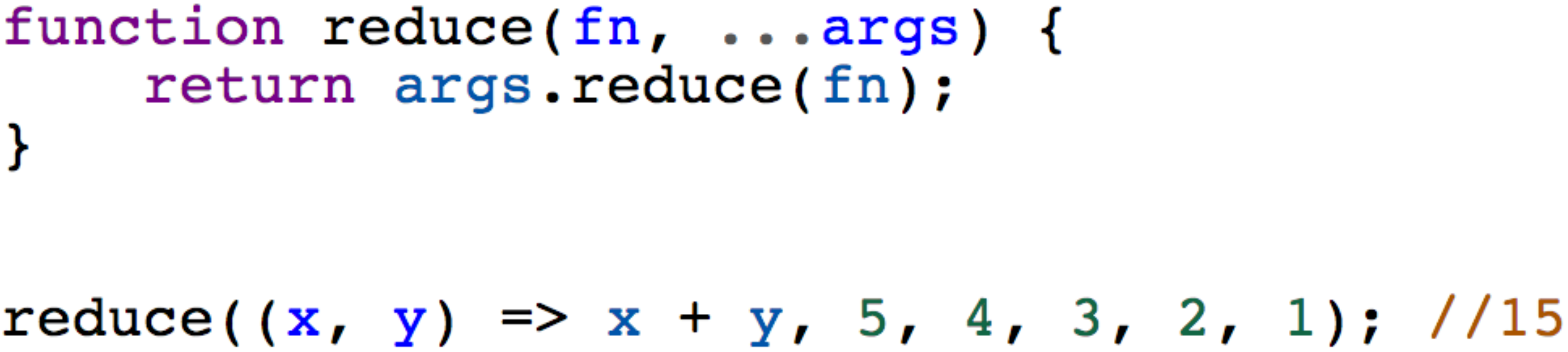
```
var pi = () => 3.1415; //no args
var sqr = a => a * a; //one arg
var add = (a, b) => a + b; //multiple args

pi(); //3.1415
sqr(5); //25
add(1, 2); //3
```

Default Params



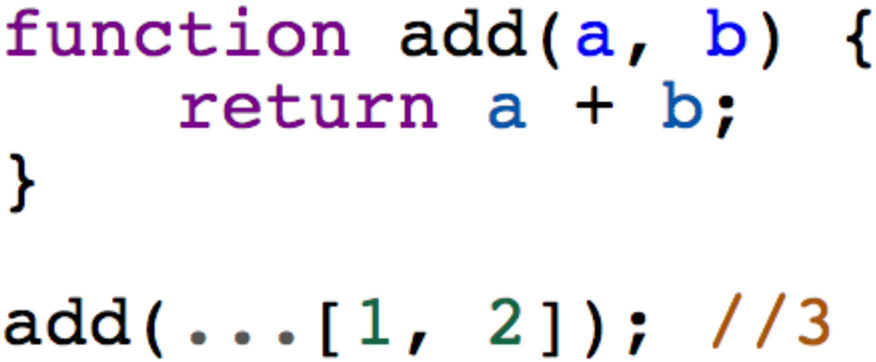
```
function getValue(val=10) {
    return val;
}

getValue(); //10
getValue(5); //5
```

Rest Params



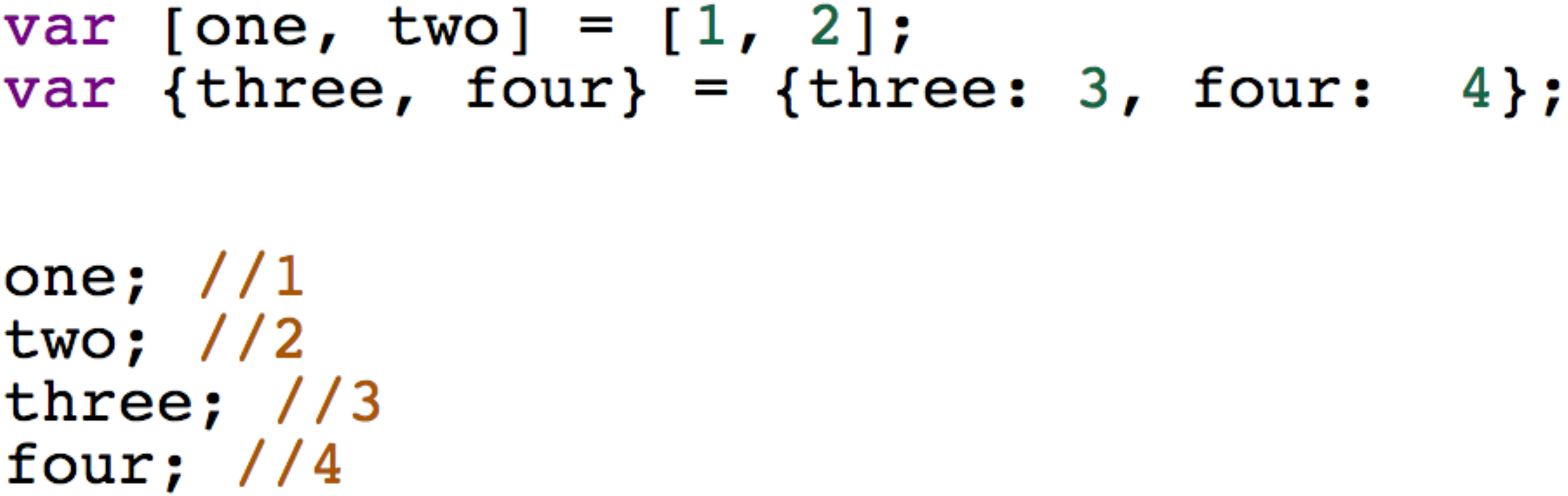
```
function reduce(fn, ...args) {
    return args.reduce(fn);
}


reduce((x, y) => x + y, 5, 4, 3, 2, 1); //15
```

Spread



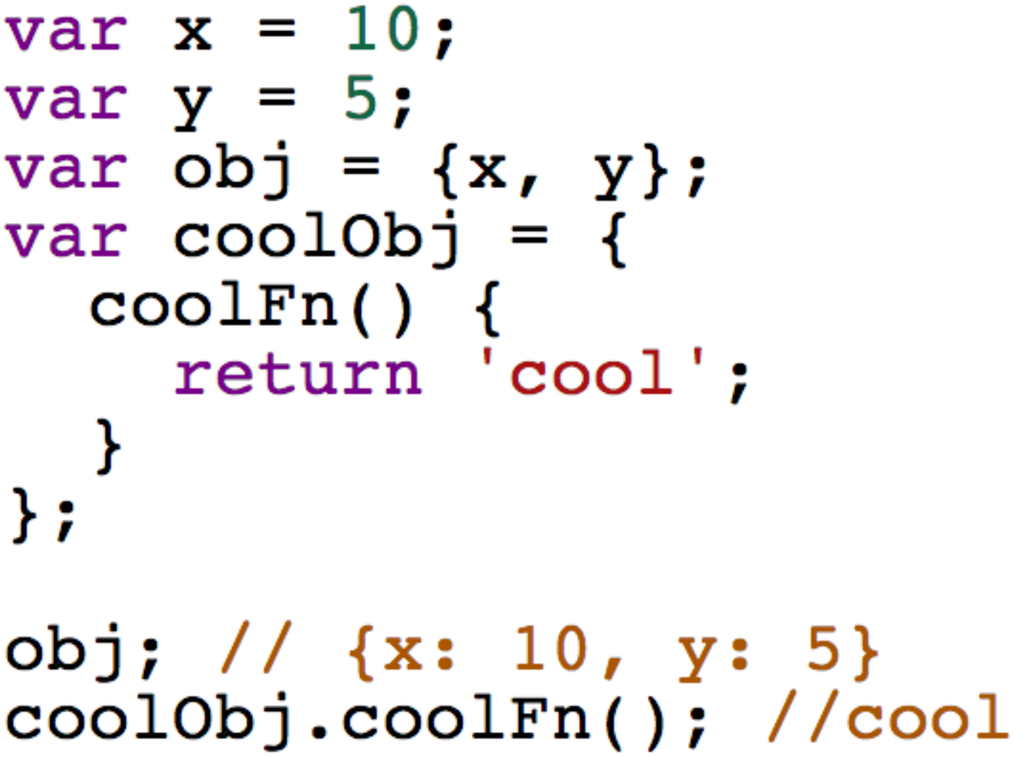
```
function add(a, b) {
    return a + b;
}

add(...[1, 2]); //3
```

Destructured Assignment



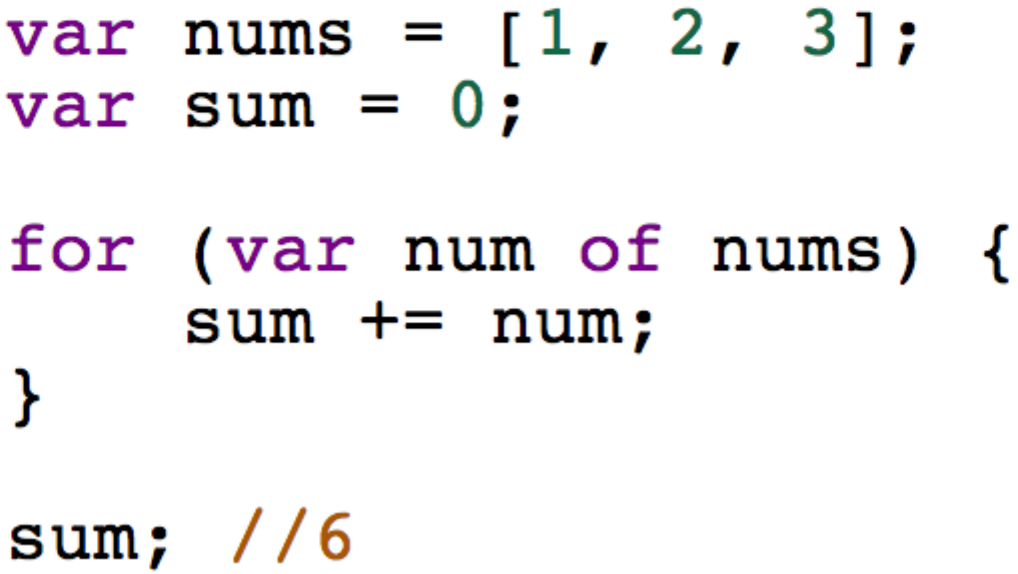
```
var [one, two] = [1, 2];
var {three, four} = {three: 3, four:  4};


one; //1
two; //2
three; //3
four; //4
```

Object Shorthand



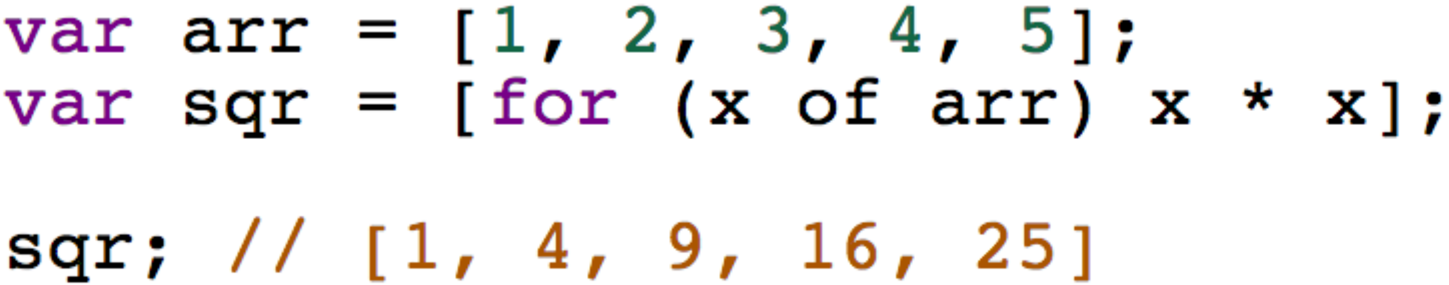
```
var x = 10;
var y = 5;
var obj = {x, y};
var coolObj = {
  coolFn() {
    return 'cool';
  }
};

obj; // {x: 10, y: 5}
coolObj.coolFn(); //cool
```

For Of



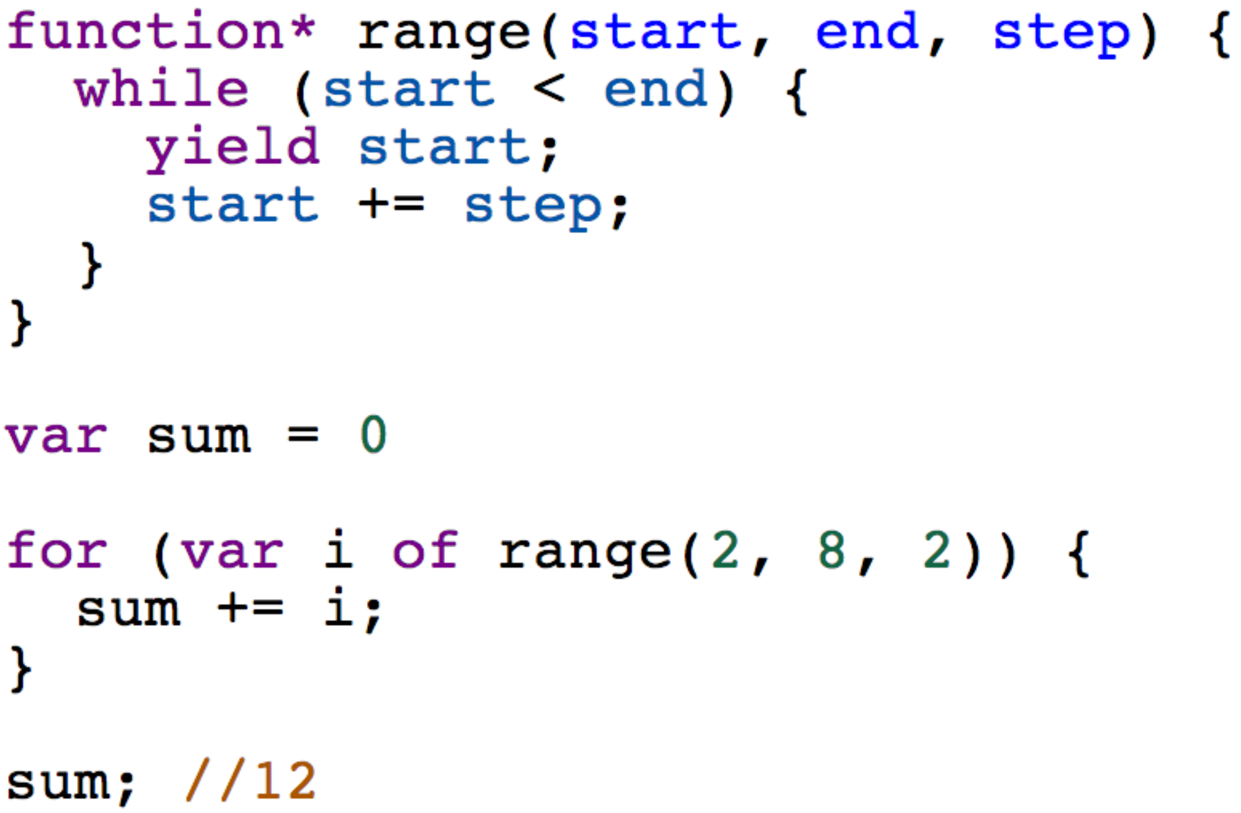
```
var nums = [1, 2, 3];
var sum = 0;

for (var num of nums) {
    sum += num;
}

sum; //6
```

Array Comprehension



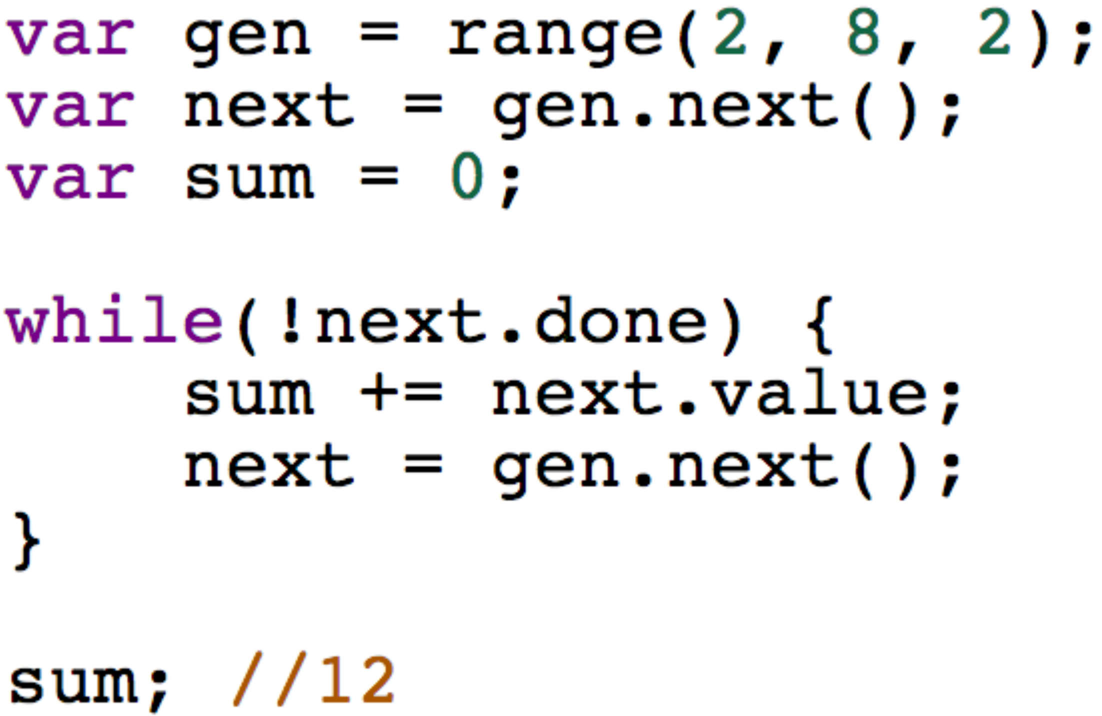
```
var arr = [1, 2, 3, 4, 5];
var sqr = [for (x of arr) x * x];

sqr; // [1, 4, 9, 16, 25]
```

Generators



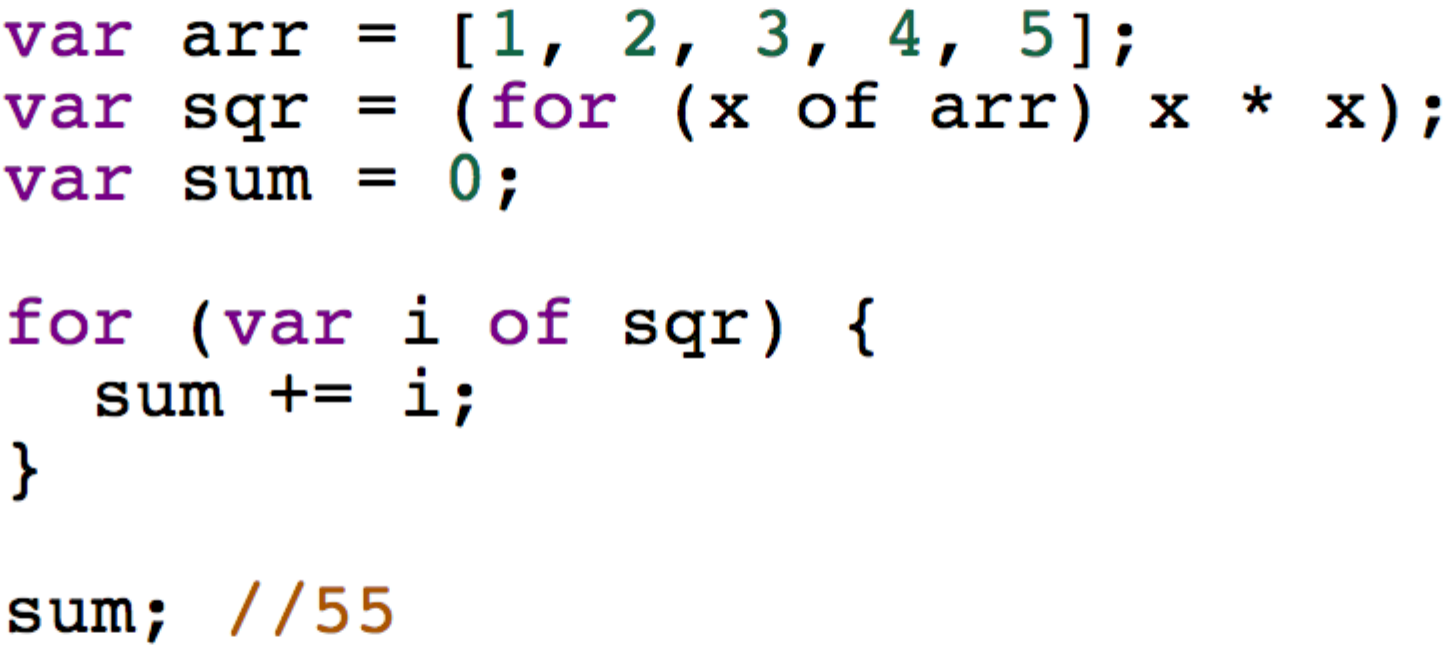
```
function* range(start, end, step) {
  while (start < end) {
    yield start;
    start += step;
  }
}

var sum = 0

for (var i of range(2, 8, 2)) {
  sum += i;
}

sum; //12
```

Generators



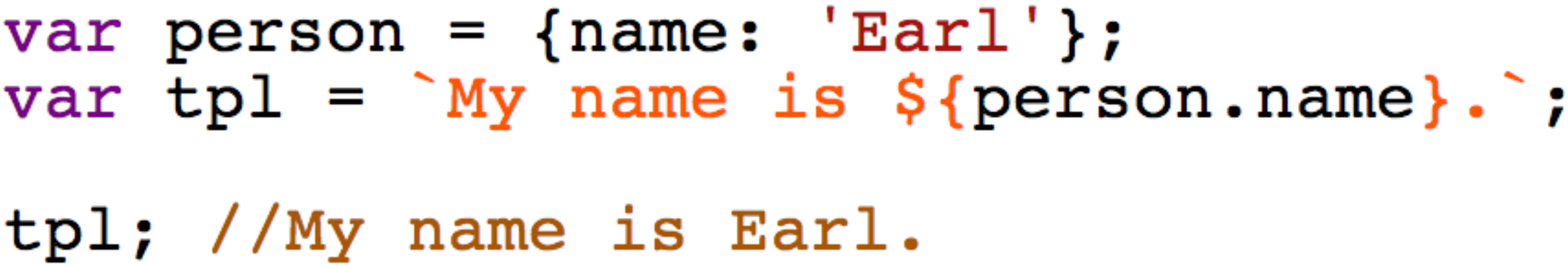
```
var gen = range(2, 8, 2);
var next = gen.next();
var sum = 0;

while(!next.done) {
    sum += next.value;
    next = gen.next();
}

sum; //12
```

Generator Comprehension



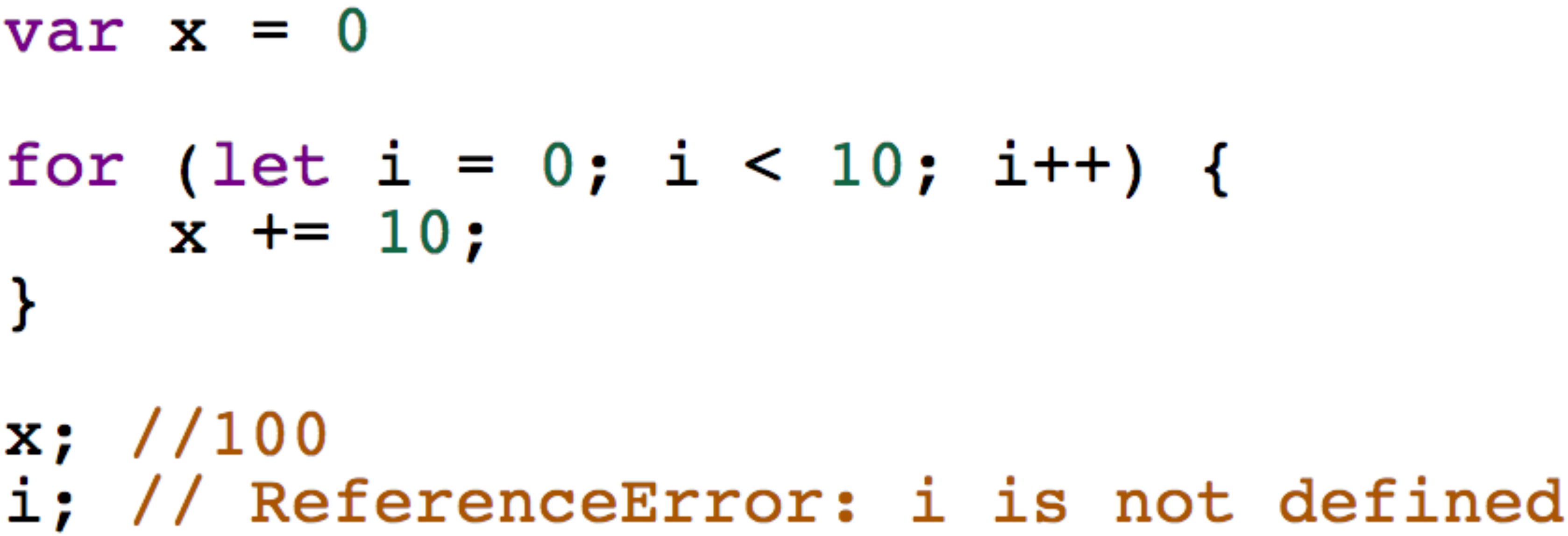
```
var arr = [1, 2, 3, 4, 5];
var sqr = (for (x of arr) x * x);
var sum = 0;

for (var i of sqr) {
  sum += i;
}

sum; //55
```

Template Literals



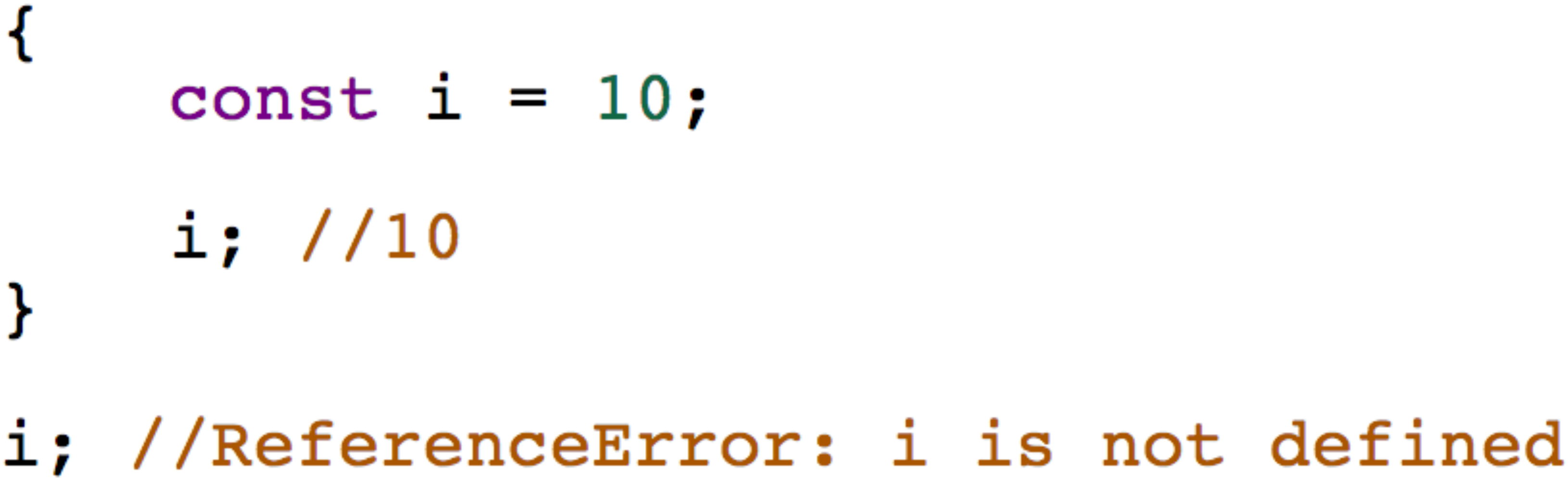
```
var person = {name: 'Earl'};
var tpl = `My name is ${person.name}.`;

tpl; //My name is Earl.
```

Block Scope



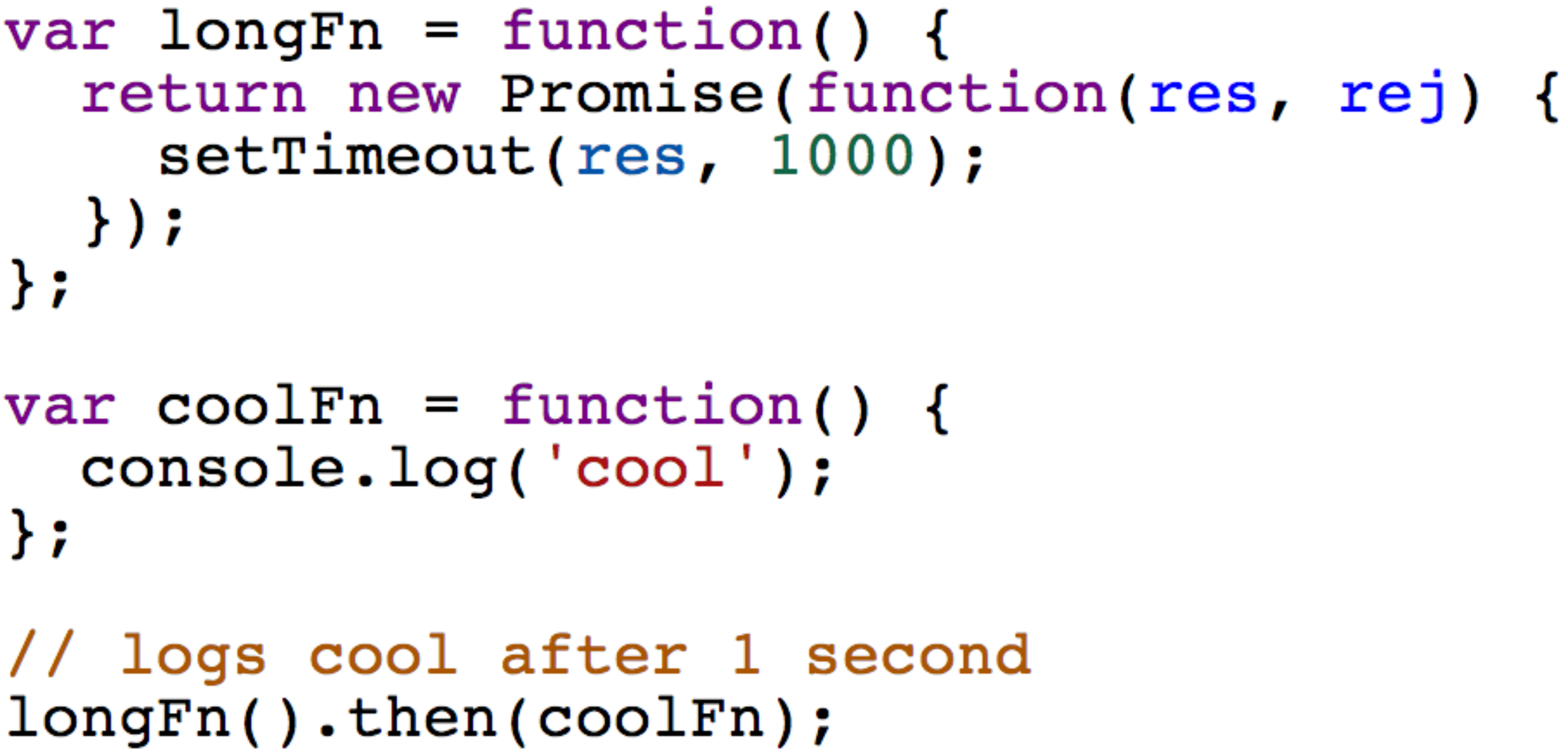
```
var x = 0

for (let i = 0; i < 10; i++) {
    x += 10;
}

x; //100
i; // ReferenceError: i is not defined
```

Block Scope



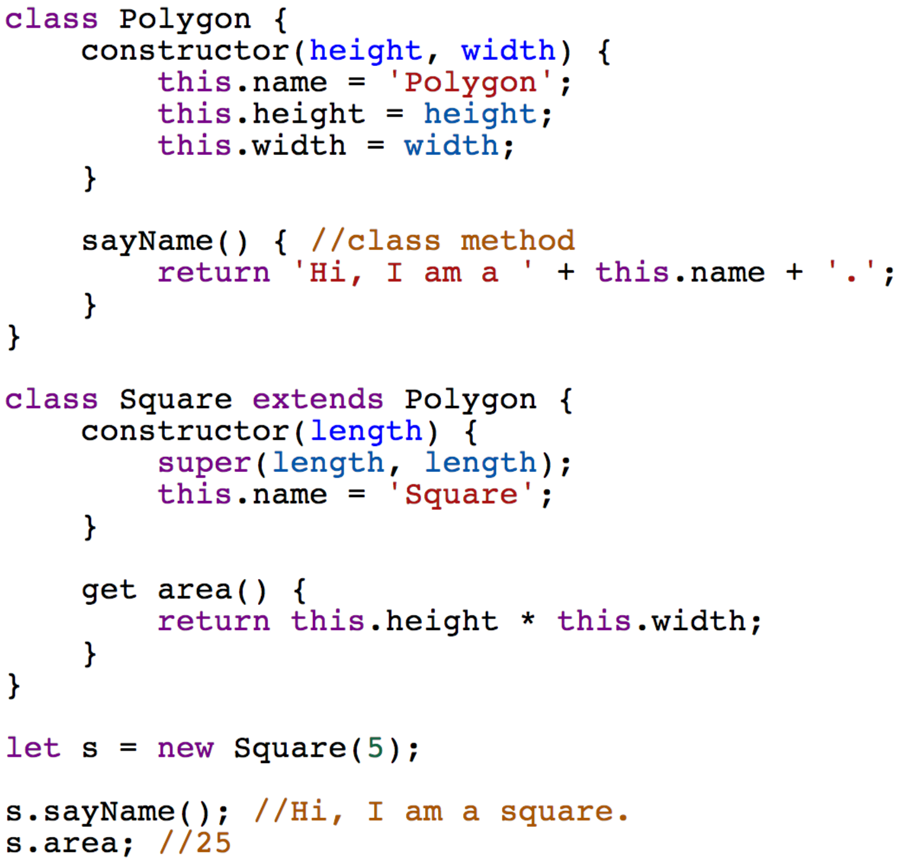
```
{
    const i = 10;

    i; //10
}

i; //ReferenceError: i is not defined
```

Promises



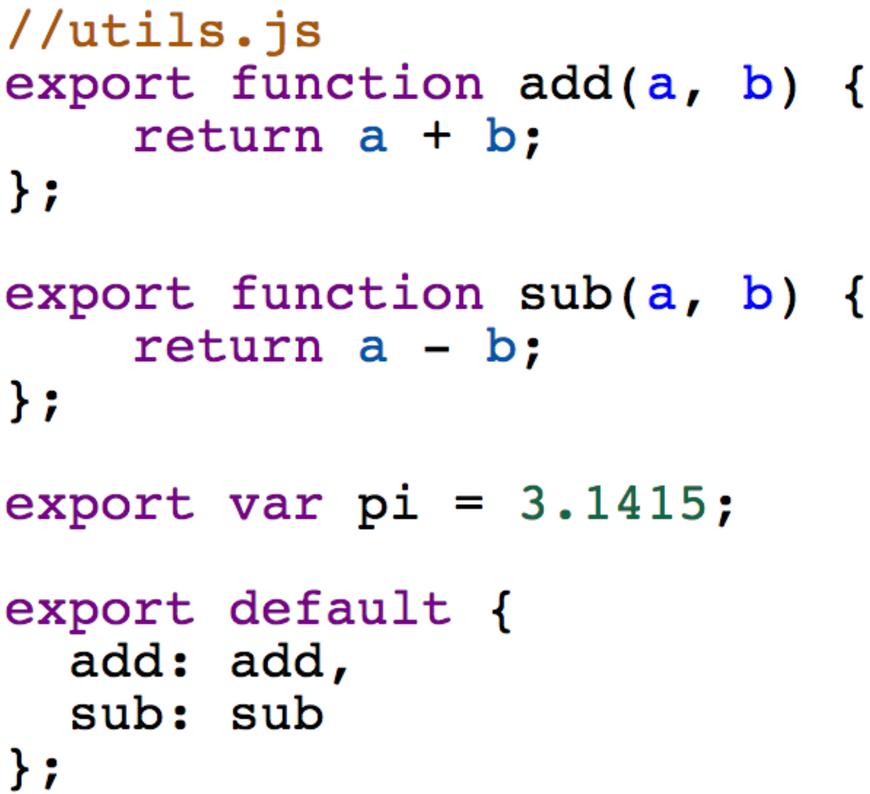
```
var longFn = function() {
  return new Promise(function(res, rej) {
    setTimeout(res, 1000);
  });
};

var coolFn = function() {
  console.log('cool');
};

// logs cool after 1 second
longFn().then(coolFn);
```

Classes



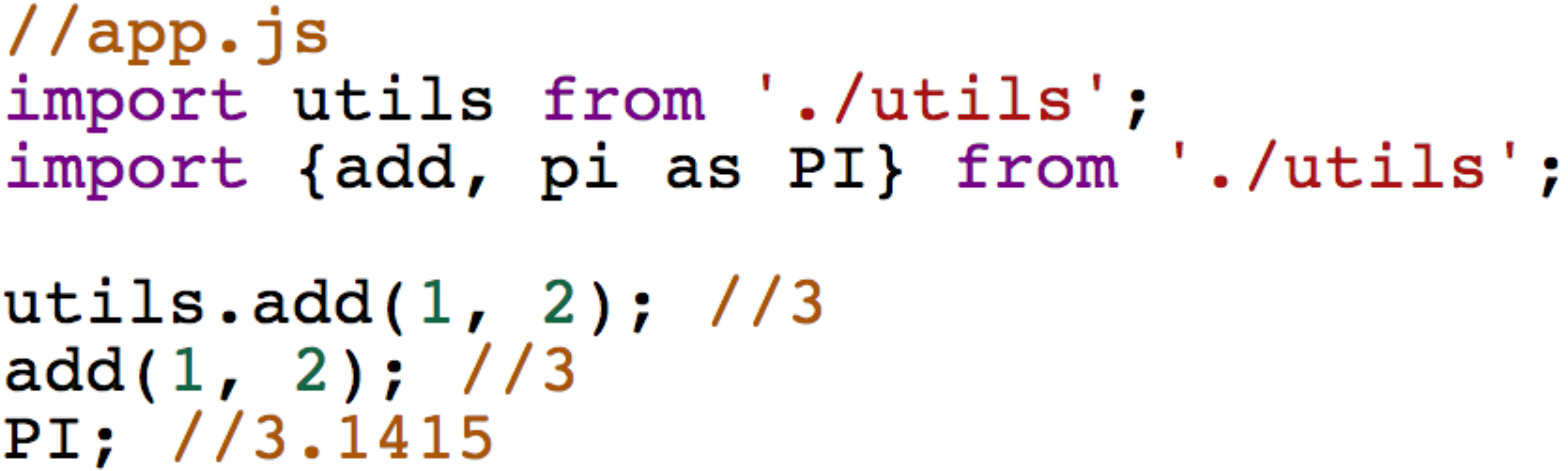
```
class Polygon {
    constructor(height, width) {
        this.name = 'Polygon';
        this.height = height;
        this.width = width;
    }

    sayName() { //class method
        return 'Hi, I am a ' + this.name + '.';
    }
}

class Square extends Polygon {
    constructor(length) {
        super(length, length);
        this.name = 'Square';
    }

    get area() {
        return this.height * this.width;
    }
}

let s = new Square(5);

s.sayName(); //Hi, I am a square.
s.area; //25
```

Modules



```
//utils.js
export function add(a, b) {
    return a + b;
};

export function sub(a, b) {
    return a - b;
};

export var pi = 3.1415;

export default {
  add: add,
  sub: sub
};
```

Modules



```
//app.js
import utils from './utils';
import {add, pi as PI} from './utils';

utils.add(1, 2); //3
add(1, 2); //3
PI; //3.1415
```

Support

- Node (--harmony)
  - 0.11 (adds several ES6 features)
- Browsers
  - Firefox > Chrome > IE > Safari
- Compilers
  - Will be compiling ES6 to ES5 for years

Compilers (Facebook)

- JSTransform (React)
  - arrow
  - class
  - some object shortcuts
  - rest params
  - template literals
- At the mercy of Esprima
  - Contributions are very low (waiting for draft?)

Compilers(Google)

- Traceur
- Can transpile offline or live
  - traceur.js vs traceur-runtime.js
  - Will be used for Angular 2.0
- Steady contribution rate
- Try it now
  - es6fiddle
  - traceur repl

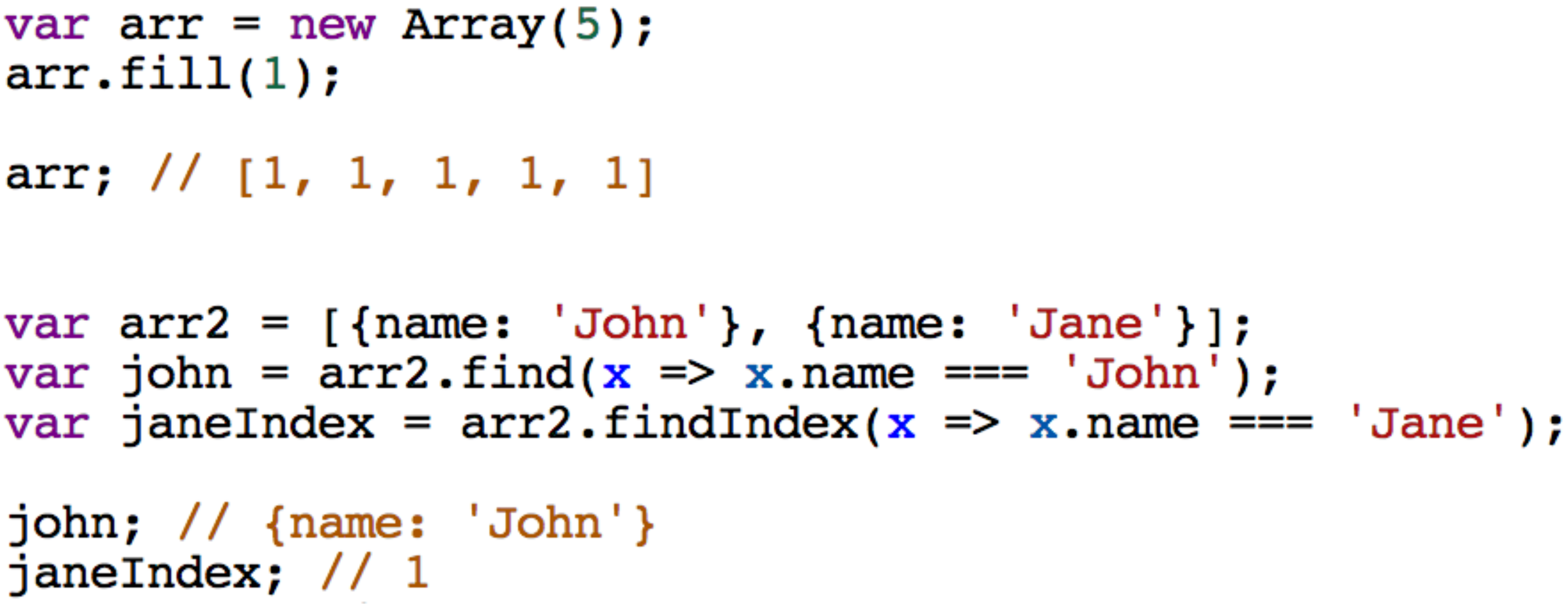
Traceur

- Most ES6 features by far
- JSHint will lint (mostly)
  - Some gotchas (2.5.1 is pretty good)
- Testable* with Jasmine + PhantomJS
  - Painful, tricky to set up
  - Coverage is tough (module loading issues)
  - Source has to be minified for PhantomJS (QtWebkit)
    - Need to try JSDom instead of PhantomJS

Traceur

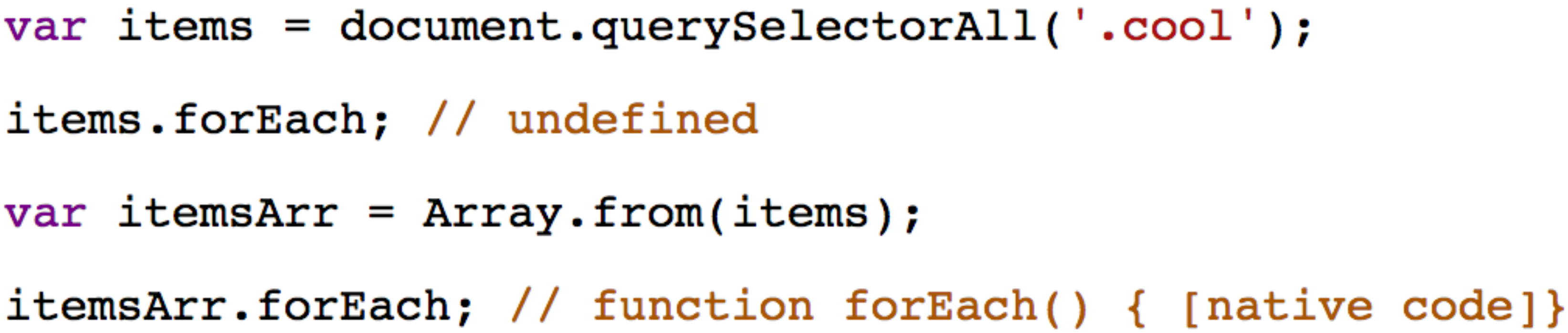
- grunt-traceur-compiler
  - Compiles for client side
  - used in Ambition production
- bower
  - traceur and traceur-runtime
- generator-esnext
  - grunt-traceur-compiler
  - grunt-contrib-jshint
  - grunt-contrib-jasmine

Traceur



```
var arr = new Array(5);
arr.fill(1);

arr; // [1, 1, 1, 1, 1]


var arr2 = [{name: 'John'}, {name: 'Jane'}];
var john = arr2.find(x => x.name === 'John');
var janeIndex = arr2.findIndex(x => x.name === 'Jane');

john; // {name: 'John'}
janeIndex; // 1
```

Traceur



```
var items = document.querySelectorAll('.cool');

items.forEach; // undefined

var itemsArr = Array.from(items);

itemsArr.forEach; // function forEach() { [native code]}
```

Future

- This time next year?
  - Spec will be published
  - Evergreen browser support
- To do
  - Experiment with ES6
  - Help build tools
- Elevate JavaScript
  - Expressive, useful
  - More people will contribute

Useful Links

- http://github.com/jmcriffey/noogajs-es6
- http://people.mozilla.org/~jorendorff/es6-draft.html
- http://github.com/google/traceur-compiler
- http://kangax.github.io/compat-table/es6/
- http://www.es6fiddle.net