

WEEK 2 DAY 2

LIGHTHOUSE  LABS

---

# ES 6 BREAKOUT

NIMA BOSCARINO

@NIMA ON SLACK



## TODAY'S TOPICS

- ▶ What is ES6? (A little bit of history)
- ▶ The Javascript Ecosystem
- ▶ "Strict" mode - from ES5
- ▶ Some new ES6 features
- ▶ Transpilers and Polyfills
- ▶ New built-in methods
- ▶ Resources

# WHAT IS ES6?

- ▶ a.k.a ES 2015
- ▶ ES => ECMAScript => JavaScript
- ▶ European Computer Manufacturers Association
- ▶ ES5 was back in 2009!
- ▶ Why do we need new standards?
  - ▶ We find better ways of expressing ourselves
  - ▶ The problems we face change over time



# JAVASCRIPT HISTORY

- ▶ Influenced by Scheme, Python, Perl...
- ▶ Netscape Navigator, 1995
- ▶ Prototype developed by Brandon Eich in 10 days
- ▶ JavaScript !== Java



# THE JAVASCRIPT ECOSYSTEM

- ▶ We'll deal with JavaScript in two places
  - ▶ Client-side: Browsers
    - ▶ Chrome (V8)
    - ▶ Firefox (Spidermonkey)
  - ▶ Server-side:
    - ▶ Node.js (V8)



**QUESTIONS?**

## “STRICT” MODE

- ▶ Introduced in ES5 (2009)
- ▶ Makes some silent errors loud
- ▶ Enforces some high-performance code
- ▶ Prohibits syntax that may be removed in future standards

```
'use strict';
```



## SOME NEW FEATURES

- ▶ For ... Of
- ▶ let & const
- ▶ String Interpolation
- ▶ Arrow Functions
- ▶ Destructuring Objects
- ▶ Spread Syntax



## FOR...OF

- ▶ For...In iterates over keys
- ▶ For...Of iterates over items
- ▶ Remember to check documentation (MDN) for specific usage!

```
> var obj = [1, 2, 3, 4]
< undefined
```

```
> for (elem in obj) {
    console.log(elem)
}
0
1
2
3
< undefined
> for (elem of obj) {
    console.log(elem)
}
1
2
3
4
< undefined
```

## LET & CONST (& VAR)

- ▶ var hoists declarations to the top of the scope
- ▶ let creates block scope variables
  - ▶ Not affected by hoisting
- ▶ const
  - ▶ Read-only reference to a value
  - ▶ Note: the value itself is not immutable, the reference just can't be reassigned
  - ▶ Global or local, depending on block

# FANCY STRING INTERPOLATION

- ▶ “Template Literals”
- ▶ String literals with embedded expressions

```
> var name = "Bruce Wayne"
< undefined

> var hero = `${name} is Batman`
< undefined

> hero
< "Bruce Wayne is Batman"
```

# QUESTIONS 2

## ELECTRIC BOOGALOO

# ARROW FUNCTIONS

- ▶ Shorter syntax than a function expression
- ▶ Does not have its own `this`
- ▶ Best suited for non-method functions (i.e. not constructors)
- ▶ No access to arguments

```
> var add = (a, b) => a + b  
< undefined  
-----  
> add(3, 4)  
< 7
```

# DESTRUCTURING OBJECTS

- ▶ Getting values out of arrays or objects one by one...
  - ▶ Annoying! Boring! Laaaaaame
- ▶ ES6: Unpacking values from arrays and objects all at once
  - ▶ Woohoo!
  - ▶ `var [first, second] = [1, 2, 3, 4, 5];`
  - ▶ `var {a, b} = { a: 1, b: 2, c: 3 }`

# SPREAD SYNTAX

- ▶ Allows an iterable or object to spread where necessary
- ▶ Iteration: taking each item of something, one after another
- ▶ Iterable: Any object that can be iterated over
  - ▶ [1, 2, 3]
  - ▶ “Hello Friends”
- ▶ Object: a set of key-value pairs
  - ▶ {name: “Shania Twain”, occupation: “musician”}



# THREE CONTEXTS FOR SPREAD SYNTAX

## ▶ 1. Function Calls

```
1 function myFunction(x, y, z) { }  
2 var args = [0, 1, 2];  
3 myFunction(...args);
```

## ▶ 2. Array Literals (and strings)

```
> var arr = [1, 2, 3]  
< undefined  
> var arr2 = [...arr, 4, 5, 6]  
< undefined  
> arr2  
< ▶ (6) [1, 2, 3, 4, 5, 6]
```

```
> [... "Tom Cruise"]  
< ▶ (10) ["T", "o", "m", " ", "C", "r", "u", "i", "s", "e"]
```

## ▶ 3. Object Literals

```
1 var obj1 = { foo: 'bar', x: 42 };  
2 var obj2 = { foo: 'baz', y: 13 };  
3  
4 var clonedObj = { ...obj1 };  
5 // Object { foo: "bar", x: 42 }  
6  
7 var mergedObj = { ...obj1, ...obj2 };  
8 // Object { foo: "baz", x: 42, y: 13 }
```

# QUESTIONS



**THIS TIME IT'S PERSONAL.**

## WHAT IF THE STANDARD ISN'T IMPLEMENTED?

- ▶ ECMA declares new features
- ▶ But that doesn't mean they get implemented everywhere!
- ▶ Transpiler
  - ▶ Write fancy new code
  - ▶ Transform to old code
- ▶ Polyfill
  - ▶ Implement new functionality with old code

## NEW BUILT-IN METHODS

- ▶ So far, we've looked at new constructs, for example...
  - ▶ For ... Of
  - ▶ Arrow Functions
- ▶ ES6 also expands the standard library
  - ▶ New string methods
  - ▶ Array Element Finding

## ES6 HAS LOTS OF NEW FEATURES...

- ▶ We can't go over them all!
- ▶ Refer to:
  - ▶ <http://es6-features.org/>
  - ▶ <https://ponyfoo.com/books/practical-modern-javascript>
  - ▶ JavaScript: The Good Parts
  - ▶ <https://caniuse.com/>

## TODAY WE LEARNED...

- ▶ ES6 is the 2015 ECMAScript standard
- ▶ Javascript runs on browsers and in Node
- ▶ “use strict” protects us from writing sloppy code
- ▶ ES6 Features:
  - ▶ For..Of, let & const, string interpolation, arrow functions
  - ▶ Destructing objects, spread syntax
- ▶ Transpilers and polyfills let us use unimplemented standards
- ▶ ES6 introduces new built-in methods `“hello”.includes(“ell”) // true`

HAPPY CODING 

LIGHTHOUSE  LABS

---

NIMA BOSCARINO

@NIMA ON SLACK