# ES6 and Unit Testing JS

# ES6: History...from Wikipedia ;)

- ECMA 262 -- Core language features
- ES 4 in 2007 -- never released by the committee
- ECMAScript 3.1 From Yahoo, MS and Google
- Eventually, ECMAScript 3.1 was standardized as ES5
- ECMAScript 2015 or Just ES6  -- feature complete in 2015
- ECMAScript 2016 or Just ES7
- ECMAScript 2017 or Just ES8

--source: https://en.wikipedia.org/wiki/ECMAScript

# Step 1: ES6

- Variables
- Functions
- Objects
- Destructuring
- Modules
- Classes and Subclassing
- Collections
- Array Capabilities

# Step 1A: Variables

- Let and const
- Block scope
- Loops and closures
- Examples

# Step 1B: Functions

- Default Parameters
- Unnamed Parameters
- Spread Operator
- Arrow Functions

# Step 1C: Objects

- Object literals
- Methods
- Computed props
- Some methods in Object

# Step 1D: Destructuring

- Destructuring objects
- Destructuring arrays
- Destructured parameters

# Step 1E: Modules

- Exporting
- Importing
- Default exports
- Importing default exports
- Import without bindings
- Modules in browsers

# Step 1F: Classes and Subclassing

- Classes in ES5
- Basic class declaration
- Class expression and named class expression
- Accessor properties
- Computed properties
- First class citizens

# Step 1G: Collections

- Sets
- Maps

# Step 1H: Array Capabilities

- Array.of & Array.from
- find and findIndex
- copyWithin

# Step 2: Browser?

- Marginal support for ES6
- Babel to the rescue!
- Use ES7
- Demo of BABEL repl

--https://babeljs.io/repl/

--https://lebab.io/try-it

# Step 3: What you got?

- Your stuff in global scope..IIFE
- External JavaScript..polluting the global scope
- No idea of dependencies
- Everything depends on everything

Identify these places and refactor

- Modularize using ES6 modules
- Create mediator/adapters for external JS
- Move libraries(underscore, Highcharts et all) to npm
- And yes, use ES6 import/export to manage dependencies

# Step 4: Testing framework

- Too many frameworks, too little time
- Testem, Jasmine, Mocha, Karma, Jest
- Jest
  - Jasmine
  - Runs in Node Environment
  - Easy mocking...once you understand it ;)
  - In built code coverage with Istanbul(html, text)
  - Can work with any assertion library
  - Out of the box with minimal conf
  - Made by Facebook to test React

# Step 5A: Write those Tests

- Pure functions--easiest

Example

```
describe('DataHelper.createListFromMap Spec', () => {
  it('is a function', () => {expect(typeof DataHelper.createListFromMap).toBe('function');});
  it('returns undefined by default', () => {expect(DataHelper.createListFromMap()).toBeUndefined();});
  it('should return the keys of map', () => {
    const map = {'Data':'Starship Enterprise', 'HAL':'United States Spacecraft Discovery One', 'TARS': 'Endurance'};
    const result = DataHelper.createListFromMap(map);
    expect(result).toHaveLength(Object.keys(map).length);
    expect(result).toContain('Data');
    expect(result).toContain('HAL');
    expect(result).toContain('TARS');
  });
});
```

# Step 5B: The Dreaded Dependencies

- Mocking needs
  - Mocking node module(fs, os, path, underscore etc)
  - Mocking functions
  - Mocking modules
- Mocking techniques
  - Mock node modules with top level __mocks__ folder
  - Mock functions with jest.fn()
  - Mock modules with a parallel __mocks__ folder

Note: Jest doesn’t merge mocked and un-mocked functions. While mocking a module, define all the functions

# Step 5B: Examples

# Standing on the shoulder of giants

- References
  - https://facebook.github.io/jest/
  - https://leanpub.com/understandinges6/read
  - https://jasmine.github.io/2.0/introduction.html
  - https://nodejs.org/api/
  - http://babeljs.io/
  - http://node.green/

# Questions?