

Modular JavaScript

JC Franco

Blake Stearman

```
console.log("hello world");
```

JC Franco

 @arfncode

 jcfranco

Blake Stearman

 @cthru

 BlakeStearman

The Story of an App

We've all been here.

- Start small
- Add features
- Grow complexity
- Add developers
- Trouble

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
</head>
<body>
</body>
</html>
```

The Story of an App

We've all been here.

- Start small
- Add features
- Grow complexity
- Add developers
- Trouble

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Stopwatch</title>
</head>
<body>
  <div>
    <div id="timer">00:00:00:000</div>
  </div>
</body>
</html>
```

The Story of an App

We've all been here.

- **Start small**
- **Add features**
- **Grow complexity**
- **Add developers**
- **Trouble**

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Stopwatch</title>
</head>
<body>
  <div>
    <div id="timer">00:00:00:000</div>
    <button type="button" id="start-button">Start</button>
    <button type="button" id="stop-button">Stop</button>
  </div>
</body>
</html>
```

The Story of an App

We've all been here.

- Start small
- Add features
- Grow complexity
- Add developers
- Trouble

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Stopwatch</title>
  <link rel="stylesheet" href="./css/app.css">
</head>
<body>
  <div class="nav">
    <a class="hoverHighlight" href="./index.html">&lt; Back</a> | Stopwatch script
  </div>
  <div class="display">
    <div id="timer">00:00:00:000</div>
    <button type="button" id="start-button">Start</button>
    <button type="button" id="stop-button">Stop</button>
  </div>
</body>
</html>
```

The Story of an App

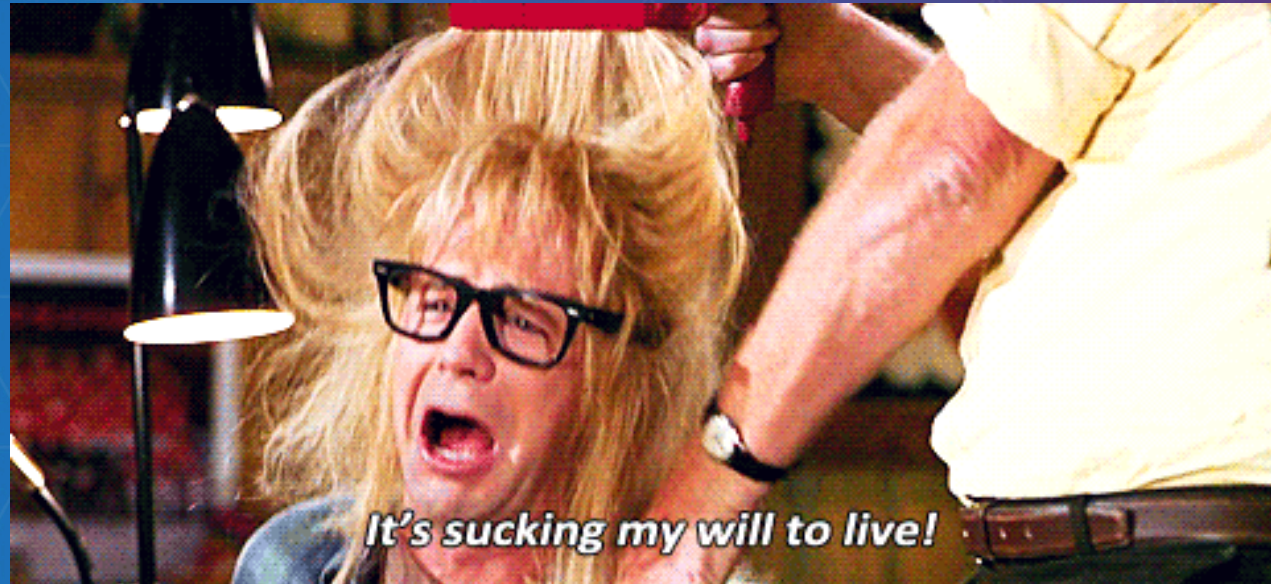
We've all been here.

- Start small
- Add features
- Grow complexity
- Add developers
- Trouble

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Stopwatch</title>
  <link rel="stylesheet" href="./css/app.css">
</head>
<body>
  <div class="nav">
    <a class="hoverHighlight" href="./index.html">&lt; Back</a> | Stopwatch script
  </div>
  <div class="display">
    <div id="timer">00:00:00:000</div>
    <button type="button" id="start-button">Start</button>
    <button type="button" id="stop-button">Stop</button>
  </div>
  <script>
    // ...
  </script>
</body>
</html>
```

Monolithic apps are

Depressing...



Modules

Building blocks of code.

- What is a module?
 - Cohesive
 - Single-purpose
 - Reusable
 - Interchangeable



- Why use modules?
 - Focused code
 - Simplifies
 - Reuse
 - Testing
 - Collaboration
 - Clean global scope
 - Promotes DRY

Don't Repeat Yourself 👍
Don't Repeat Yourself 👍
Don't Repeat Yourself 👍

JavaScript Module Pattern

The foundation

- **Immediately-Invoked Function Expression (IIFE)**
 - Truly private scope
 - Executes immediately
 - Can return value

```
(function () {  
  // your awesome code ✨ 🌟  
})();
```

Diving Into the Pattern

- **Creates namespace**
- **Private data**
- **Expose API**

```
var MyModule = (function () {  
  var myData = 0; // private data  
  
  function myPrivateMethod () {  
    // private - not exposed through API  
  };  
  
  function myPublicMethod () {  
    // public - exposed through API  
  };  
  
  // MyModule API  
  return {  
    publicMethod: myPublicMethod  
  };  
})();
```

Now stop... Demo Time!



Module Pattern Caveats

- Name collisions
- Loading order matters
- Truly private data
 - Limits extension

```
// rewire Stopwatch.displayTime() to return raw time
Stopwatch.displayTime = function () {
  return Stopwatch._time(); // _time() is private
};
```

```
Stopwatch.displayTime(); // TypeError: undefined is not a function
```

Variation on Pattern

To make our lives easier.

- **Expose all**
- **Private as convention**
- **Improves**
 - Debugging
 - Extension

```
var MyModule = (function () {  
  // MyModule API  
  return {  
    // private  
    _data: 0,  
    _privateMethod: function () { },  
  
    // public  
    publicMethod: function () { }  
  };  
})();
```

Module Definitions

Taking the module pattern to another level.

- **Asynchronous Module Definition (AMD)**
- **CommonJS**
- **Universal Module Definition (UMD)**
- **ECMAScript 6 Harmony**

Asynchronous Module Definition (AMD)

- Asynchronous loading
- Web-based solution
- Lazy loading
- Implementation
 - define
 - require

```
// moduleA.js
define(["moduleB"], function (moduleB) {
  // module API
  return {
    _data: 100,
    calculate: function () {
      moduleB.calculate(this._data);
    }
  };
});

// main.js
require(["moduleA"], function (moduleA) {
  moduleA.calculate();
});
```


AMD

```
// moduleA.js
define(["moduleB"], function (moduleB) {
  // module API
  return {
    _data: 100,
    calculate: function () {
      moduleB.calculate(this._data);
    }
  };
});
```

Look familiar?

```
var moduleA = (function (moduleB) {
  // module API
  return {
    _data: 100,
    calculate: function () {
      return moduleB.calculate(this._data);
    }
  };
})(moduleB);
```



CommonJS

- Desktop environment*
- Synchronous
- Can only export objects*

```
// moduleA.js
var moduleB = require("moduleB");

var _data = 100;

// module API (exports)
exports.calculate = function() {
  moduleB.calculate(_data);
};

// main.js
var moduleA = require("moduleA");

moduleA.calculate();
```

UMD

- Supports browser and desktop
- Boilerplate
- Variations

```
// moduleA.js
(function () {
  define(["moduleB"], function (moduleB) {...});
})(
  // AMD or CommonJS environment?
  typeof define == "function" && define.amd
  ? define
  : function (ids, factory) {
    var deps = ids.map(function (id) { return require(id); });
    module.exports = factory.apply(null, deps);
  }
);
```

```
define(["moduleB"], function (moduleB) {
  return function () {
    return {
      _data: 100,
      calculate: function () {
        moduleB.calculate(this._data);
      }
    };
  };
});
```

ES6 Harmony

- ECMAScript standard
- Supports browser and desktop
- Coming to an interpreter near you



```
// moduleA.js
import moduleB from "moduleB";

var _data = 100;

export function calculate () {
  return moduleB.calculate(_data);
}

// main.js
import moduleA from "moduleA";

moduleA.calculate();
```

Frameworks

The good, the bad... and the ugly.

The Good

- **Structured**
- **Modular**
- **Scaffolding**
- **Documented (usually)**

The Bad

- **Overhead**
- **Bugs**
- **Vulnerabilities**

The Ugly

- **Mixing**
- **Interchanging**

Dojo Toolkit

<http://dojotoolkit.org>

- **Foundation of ArcGIS JavaScript API**
- **AMD**
- **Class-based inheritance**
- **Internationalization**

dojō
toolkit

Why Dojo? https://developers.arcgis.com/javascript/jshelp/why_dojo.html

Defining Modules in Dojo

- **define**
- **declare***

```
define(["dojo/_base/declare"], function (declare) {  
  return declare(null, {  
    _data: 100,  
    getData: function () {  
      return this._data;  
    }  
  });  
});
```

Defining Widgets in Dojo

- Extend `_WidgetBase`

```
define([
  "dojo/_base/declare",
  "dijit/_WidgetBase",
], function (declare, _WidgetBase) {
  return declare([_WidgetBase], {
    declaredClass: "MyWidget",
    _data: 100,
    getData: function () {
      return this._data;
    }
  });
});
```


Dojo Widget

What do you gain?

- **Lifecycle**



constructor

postMixInProperties

buildRendering

postCreate

startup

destroy

Dojo Templated Widget

Putting a face on it.

- **Extend `_TemplatedMixin`**

```
define([
  "dojo/_base/declare",
  "dijit/_WidgetBase",
  "dijit/_TemplatedMixin"
], function (declare, _WidgetBase, _TemplatedMixin) {

  return declare([_WidgetBase, _TemplatedMixin], {
    declaredClass: "MyHelloWorld",
    templateString: "<div><label>Hello World!</label></div>"
  });

});
```


Dojo Templated Widget

Continued...

- **Best practice** 
 - define template in separate file
 - import template file as a string


```
// ./MyHelloWorld.js
define([
  "dojo/_base/declare",
  "dijit/_WidgetBase",
  "dijit/_TemplatedMixin",
  "dojo/text!./templates/MyHelloWorld.html"
], function (declare, _WidgetBase, _TemplatedMixin, template) {
  return declare([_WidgetBase, _TemplatedMixin, template], {
    declaredClass: "MyHelloWorld",
    templateString: template
  });
});
```

```
<!-- ./templates/MyHelloWorld.html -->
<div>
  <label>Hello World!</label>
</div>
```



Styling a Templated Widget

Making it look pretty.

- Use CSS
- Inline CSS?
 - Styling bound with template
 - Harder to fix styling issues
- Best practice 
 - Define separate CSS file
 - Namespace using class name
 - Inject namespace class name
 - Import your CSS file

```
<div>  
  <label style="font-size: 24px;  
    color: chartreuse;">  
    Hello World!  
  </label>  
</div>
```



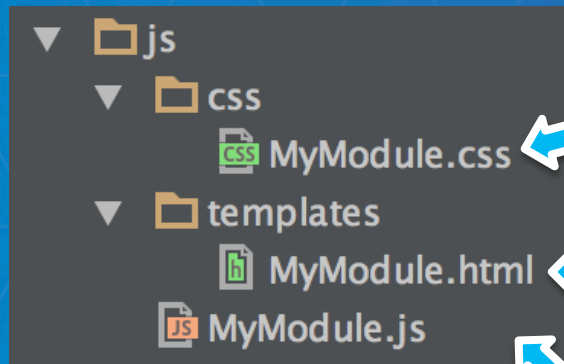
```
<div class="myHelloWorld">  
  <label>Hello World!</label>  
</div>
```

```
.myHelloWorld label {  
  font-weight: bold;  
}
```



Styling Best Practice

What should this look like?



```
/* ./css/MyHelloWorld.css */  
.myHelloWorld label {  
  font-weight: bold;  
}
```

```
<!-- ./templates/MyHelloWorld.html -->  
<div>  
  <label>Hello World!</label>  
</div>
```

```
// ./MyHelloWorld.js  
define(  
  "dojo/_base/declare",  
  "dijit/_WidgetBase",  
  "dijit/_TemplatedMixin",  
  "dojo/text!./templates/MyHelloWorld.html"  
) , function (declare, _WidgetBase, _TemplatedMixin, template) {  
  
  return declare("MyHelloWorld", [_WidgetBase, _TemplatedMixin, template], {  
    baseClass: "myHelloWorld",  
    templateString: template  
  });  
};
```

48:15:16:2342

Start

Stop

Revisiting Stopwatch

Dojo Conversion

Recap

Things to remember.

- **Modular JavaScript**
 - Break code into cohesive, reusable modules
 - Each has specific purpose
 - Each contains code only for that purpose
- **Best practices**
 - DRY
 - Separate files (code, template, styles)
 - Import templates using `dojo/text`
 - Namespace styles
- **Dojo/Dijit Modules**
 - `dijit/_WidgetBase`
 - `dijit/_TemplatedMixin`
 - `dijit/_WidgetsInTemplateMixin`
 - `dijit/a11yclick`
 - `dojo/Evented`
 - `dojo/on`
 - ...

Resources

Online swag.

- **Stopwatch demo files**

- <https://github.com/BlakeStearman/stopwatch-devsummit-2015>

- **JavaScript Web Optimizer**

- <https://jso.arcgis.com/>

- **Dijit**

- <http://dojotoolkit.org/reference-guide/1.10/dijit/>

- <http://dojotoolkit.org/reference-guide/1.10/quickstart/writingWidgets.html>

- https://dojotoolkit.org/documentation/tutorials/1.10/recipes/custom_widget/

- **Books**

- [Learning JavaScript Design Patterns by Addy Osmani](#)

- [JavaScript Patterns by Stoyan Stefanov](#)

Rate This Session

www.esri.com/RateMyDevSummitSession

