



Esri International Developer Summit
Palm Springs, CA

Modular JavaScript

Derek Swingley

@derekswingley

github.com/swingley

Matt Driscoll

@driskull

github.com/driskull

Agenda

- Why?
- OOP in JS
- Vanilla JS Classes
- Modules, AMD and Dojo
- Widgets
- Walkthrough

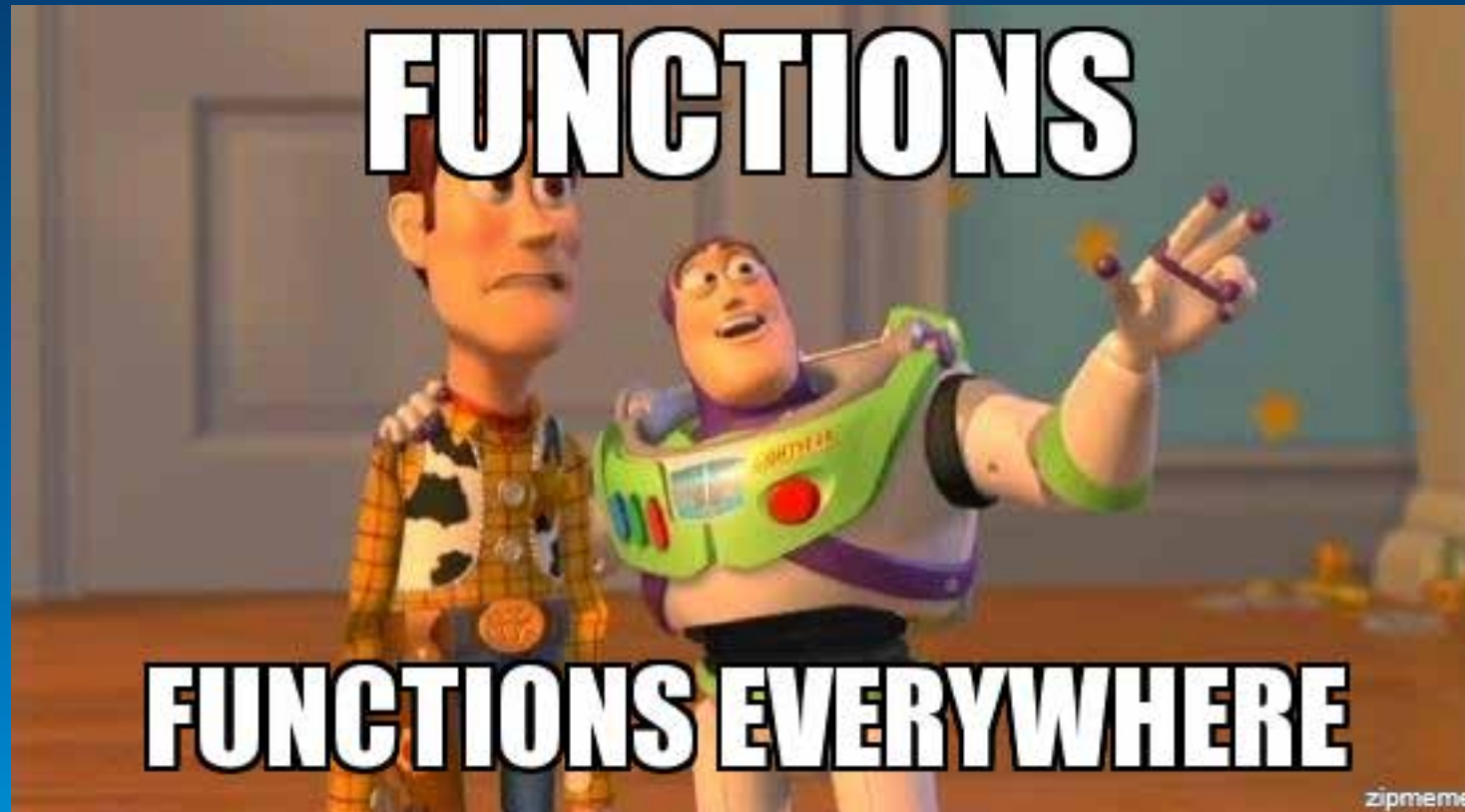
Classes?
Modules?
Why?

Object Oriented

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism



JavaScript has no class



Let's make a Class

But...

- Scope?
- Multiple inheritance?
- Dependency management?

AMD Modules

dojo
toolkit

ArcGIS JS API Docs

Returns the Map Class as a parameter

Class: Map

[AMD Module Require | Legacy Module Require]

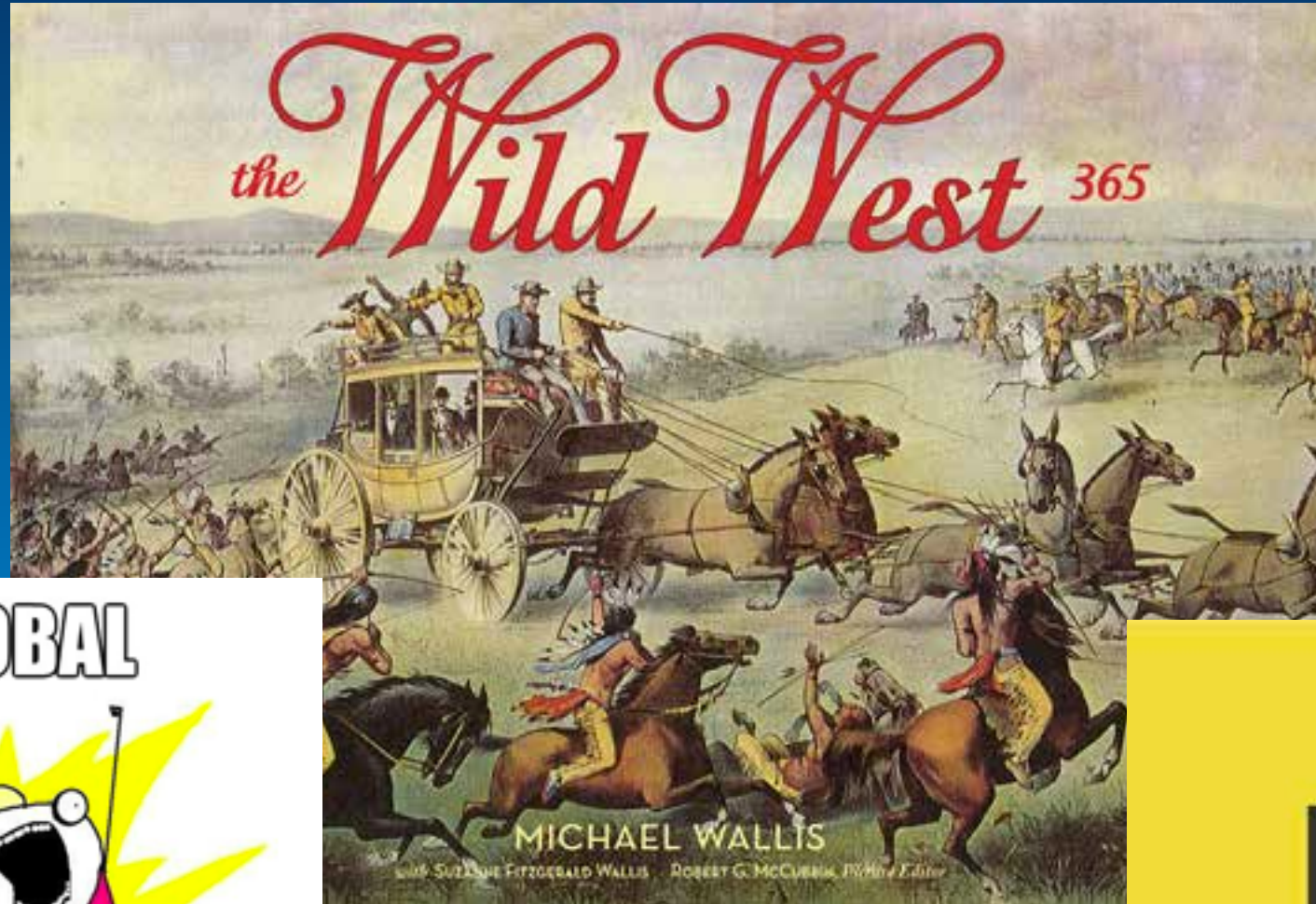
```
require(["esri/map"], function(Map) { /* code goes here */ });
```

Class: Map

[AMD Module Require | Legacy Module Require]

```
dojo.require("esri.map");
```

Before AMD



Asynchronous Module Definition (AMD)

- Fewer globals
- Plays nice
 - can be used with other AMD compliant code
- Improvements
 - Asynchronous loading
 - Dependency handling
 - Cross origin loading



Dojo Loader

- **define()**
 - Creates a module

```
1 require(["someDependency"], function(someDependency){
2     var myModule = someDep.doSomething();
3     return myModule;
4 });
5
```

- **require()**
 - Gets a module

```
1 require(["someModule"], function(someModule){
2     someModule.doSomething();
3 });
4
```

Using Custom Modules

```
1 <script type="text/javascript">
2   var dojoConfig = {
3     packages: [{
4       name: "application",
5       location: "/js"
6     }, {
7       name: "config",
8       location: "/config"
9     }]
10  };
11 </script>
12
```

Custom Name

URL Path

Dojo Config HTML5 Data Attribute

```
1 <script src="//js.arcgis.com/3.9" data-dojo-config="packages:  
  [{name:'application',location:'/js'},{name:  
  'config',location:'/config'}]"></script>
```

2

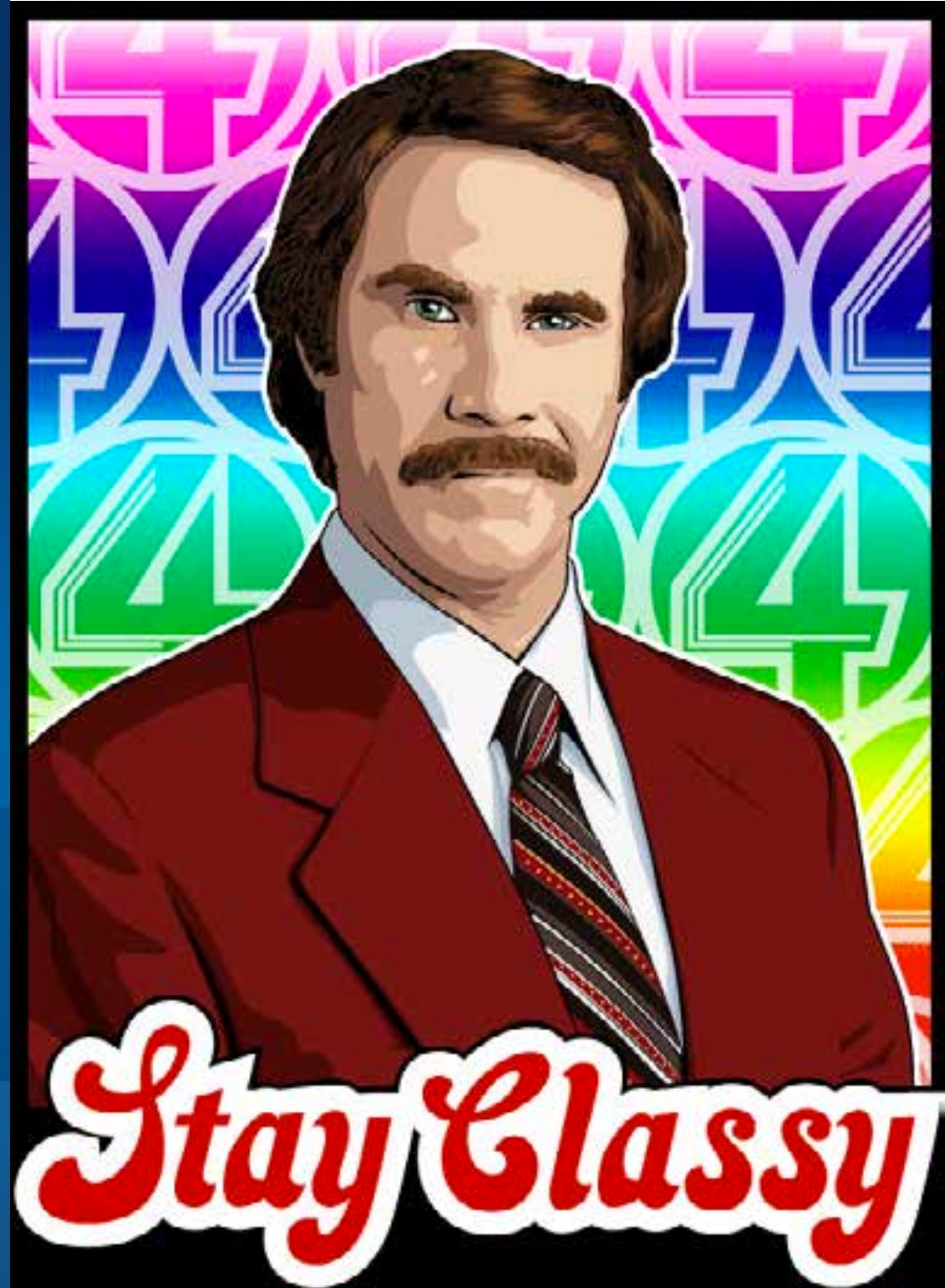
A Simple AMD Module

```
1 // config/myConfig.js
2 define({
3     "appid": "",
4     "webmap": "de5ae0c2040c49d38e9ea0637454ac73",
5     "proxyurl": "",
6     "theme": "blue"
7 });
8
```


Using The Simple Module

```
1 require(["config/myConfig"], function(myConfig){
2     console.log(myConfig.webmap);
3 });
4
```


AMD Classes



Writing Classes

- Use “`dojo/_base/declare`” to create a class
- `declare(className, superclass, props);`

Parameter	Type	Description
<code>className</code>	String	The optional name of the constructor
<code>superclass</code>	Function Function[]	May be null, a Function, or an Array of Functions.
<code>props</code>	Object	An object whose properties are copied to the created prototype

Simple Classes

```
1 // Anonymous class
2 define(["dojo/_base/declare"], function(declare){
3     return declare(null, {
4         // Custom properties and methods here
5     });
6 });
7
```

Inherited Classes
(Optional)
["esri/Map", "my/Module",...]

```
1 // Named class
2 define(["dojo/_base/declare"], function(declare){
3     return declare("mynamespace.MyClass", null, {
4         // Custom properties and methods here
5     });
6 });
7
```

Global Namespace
(Optional)

* Named classes should only be created if they will be used with the Dojo parser.

AMD Widgets



Dijits Are Classy Too

- **Templates**
- **Common lifecycles**
- **Events**
 - **Owning event handles**
 - **Custom widget events**
- **Properties that can be**
 - **Get**
 - **Set**
 - **Watched**



"dijit/_WidgetBase"

- Foundation class.
- Inherit this class to create a widget.
- Handles widget lifecycle methods.



"dijit/_TemplatedMixin"

- Provide a string of HTML for the widget to use as a template
- String may have
 - Variable Substitution - `${property}`
 - Attach points for nodes - `data-dojo-attach-point`

"theme" Property of widget

Reference to this node

```
1 <div class="${theme}" role="presentation">
2   <div class="${_css.container}">
3     <div data-dojo-attach-point="_homeNode"
4     title="${_i18n.widgets.homeButton.home.title}" role="button" class="${_css.home}">
5     <span>${_i18n.widgets.homeButton.home.button}</span></div>
6   </div>
</div>
```

Localized text

"dojo/Evented"

- Base class allowing a developer to emit events
- Users can listen with “dojo/on”

```
1 this.emit("find", { "results": myResults });  
2
```

```
1 myWidget.on('find', function(response){  
2     var results = response.results;  
3 });
```


"dojo/Stateful"

- Base class for getting, setting, and watching for property changes
- Inherited by "dijit/_WidgetBase"

```
1 myWidget.get("myProperty");  
2  
3 myWidget.set("myProperty", "new value");  
4  
5 myWidget.watch("myProperty", doThisFunction);  
6
```

Walk-through Example

<https://github.com/driskull/arcgis-dijit-home-button-js>



Tips

- Refer back to the latest Dojo API
- Some functions deprecated or replaced
 - Use newer Dojo functions if available
- Use new event handling with `on()`
- Get/Set widget properties with `.get()` and `.set()`
- `watch()` widget properties and call update functions



Session survey:
<http://esriurl.com/7517>



Understanding our world.

Derek Swingley
@derekswingley

Matt Driscoll
@driskull