Towards Pasta Code Nirvana

Using JavaScript MV* to Fill Your Programming Ravioli

Bret Davidson | NCSU Libraries

Metaphors for describing software architectures.



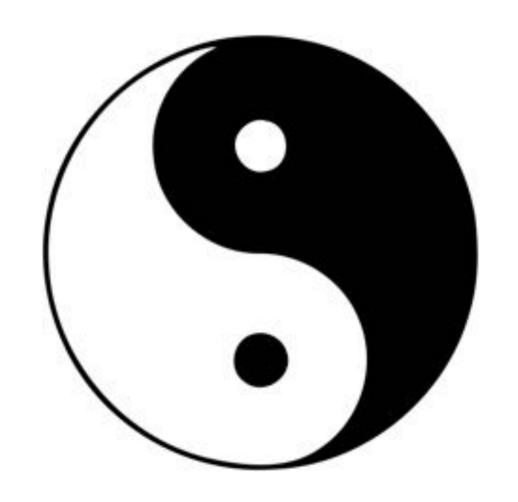








Modular Reusable Testable



Why JavaScript?

Brief History of JavaScript

- Created by Brendan Eich in 1995
- IE support in 1996
- Standardized by ECMA in 1997
- JJG coins AJAX in 2005
- V8 JS engine in 2008
- JS MVC circa 2010

JS is everywhere

JS is fast(er)

JS is the lingua franca

What is JS MV*?

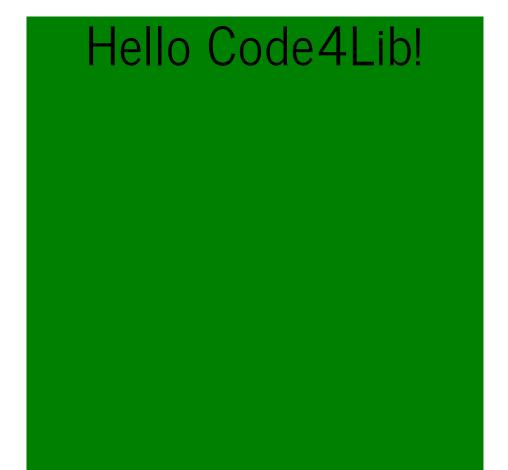
MVC - MVP - MVVM

Approximate Architecture

- Models
- Views
- Controllers
- Data Binding

Hello Code4Lib!

Change MSG



```
function InputCtrl ($scope) {
   $scope.data = 'Hello Code4Lib!';

   $scope.updateMessage = function (msg) {
     $scope.data = msg;
   };
}
```

MV* Benefits

- Abstracts DOM interaction
- Encourages code reuse
- Enforces modularity
- Improves testability

Architecture that supports best practices.

When to use?

- JS heavy applications
- Interactive applications
- Distributed applications
- Modern devices/browsers

Who is using MV*?

- USA Today
- Rdio
- Hulu
- Discourse
- Youtube on PS3
- General Motors

When to avoid?

- Need to support no-JS users
- App is document driven (blog)
- Older devices/browsers

Frameworks

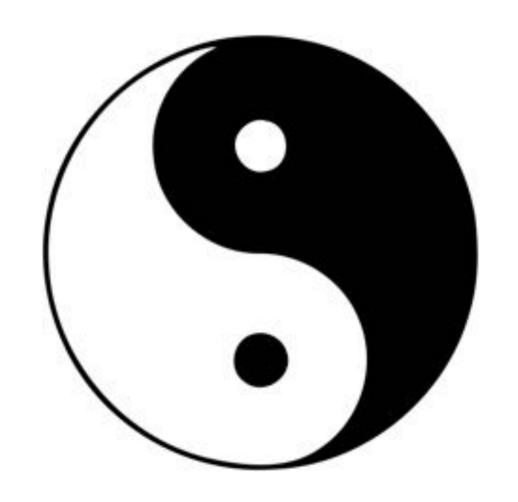
- Angular
- Ember
- Backbone
- Many Others

MV* Throwdown

http://todomvc.com/

My Experiences

Suma github.com/cazzerson/Suma



Community

Loose Coupling

Cognitive Load

