Introduction to

ANGULARJS

by Google

Which Version of Angular?

Which Version of Angular?

We’re covering Angular 1 NOT Angular 2

Which Version of Angular?

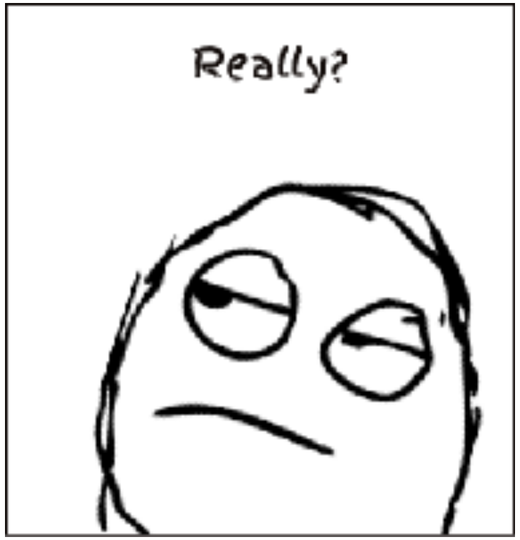
We’re covering Angular 1 NOT Angular 2

…because they’re so different as to nearly be different frameworks

Which Version of Angular?

We're covering Angular 1 NOT Angular 2

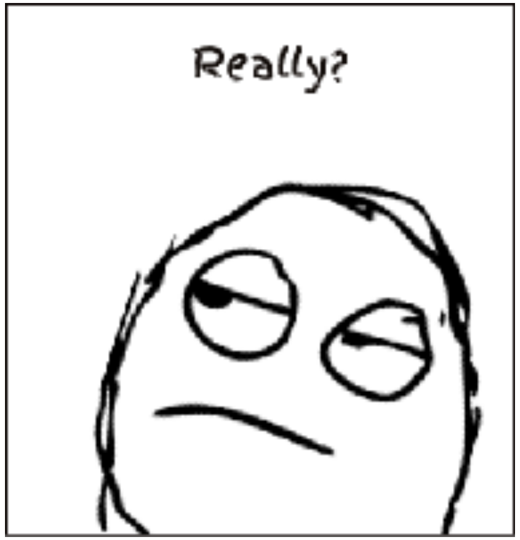
...because they're so different as to nearly be different frameworks



Which Version of Angular?

We're covering Angular 1 NOT Angular 2

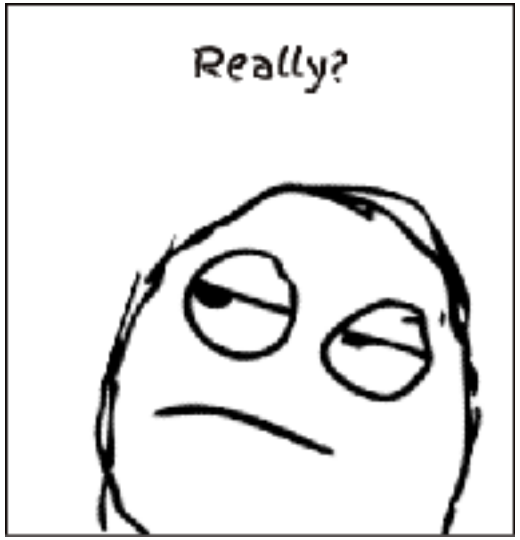
...because they're so different as to nearly be different frameworks

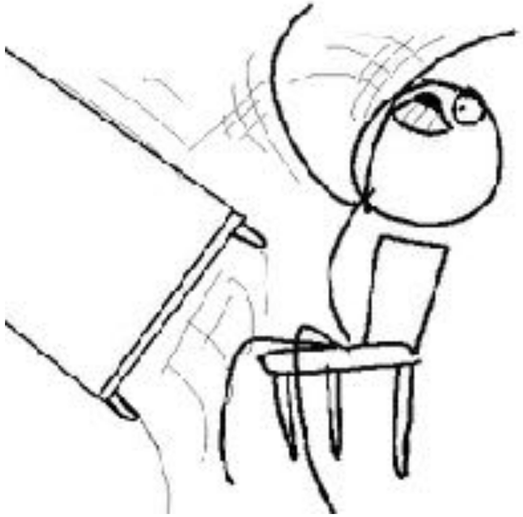


Which Version of Angular?

We're covering Angular 1 NOT Angular 2

…because they're so different as to nearly be different frameworks





Agenda

Agenda

1. What is Angular?

Agenda

1. What is Angular?
2. Pros & cons

Agenda

1. What is Angular?
2. Pros & cons
3. Quick start guide

Agenda

1. What is Angular?
2. Pros & cons
3. Quick start guide
4. Style guide

Agenda

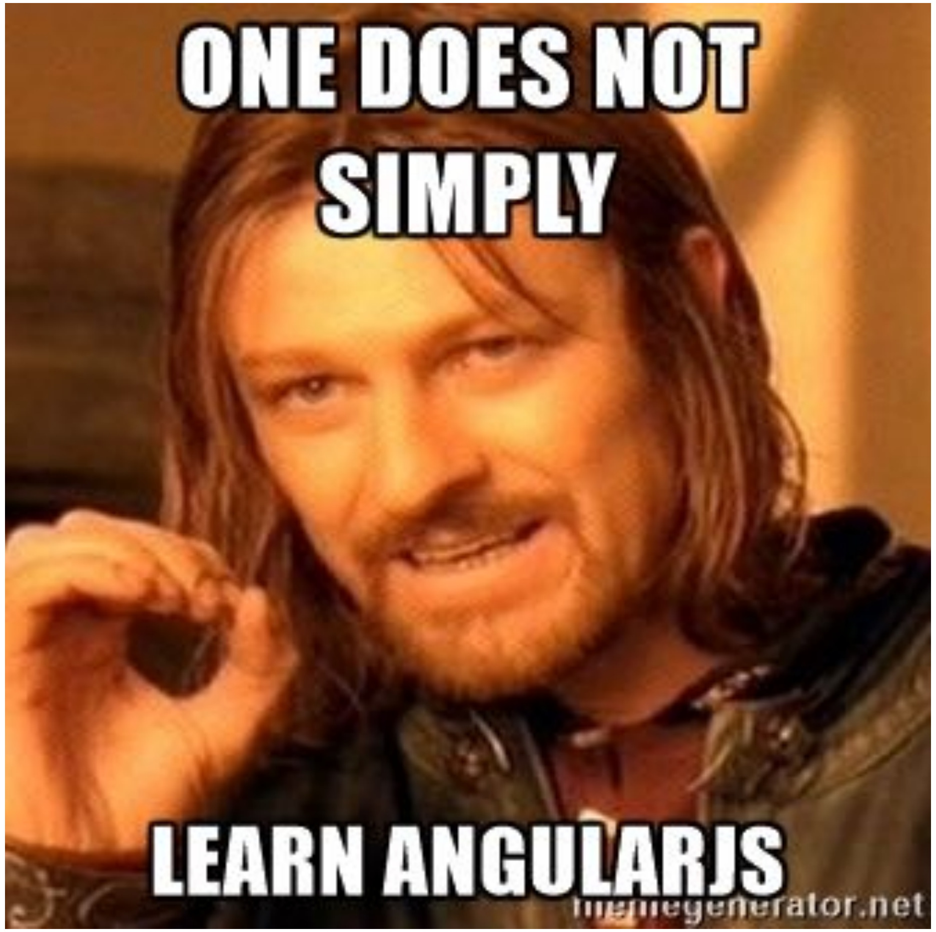
1. What is Angular?
2. Pros & cons
3. Quick start guide
4. Style guide
5. Build a to-do list

Agenda

1. What is Angular?
2. Pros & cons
3. Quick start guide
4. Style guide
5. Build a to-do list
6. Q&A

Agenda

1. What is Angular?
2. Pros & cons
3. Quick start guide
4. Style guide
5. Build a to-do list
6. Q&A



What is Angular?

Angular is an open source, client-side framework that extends HTML using JavaScript. It uses the MVC (model-view-controller) approach to design. Angular is designed for building single page applications (SPA).

What is Angular?

Angular is an open source, client-side framework that extends HTML using JavaScript. It uses the MVC (model-view-controller) approach to design. Angular is designed for building single page applications (SPA).

What is Angular?

software with source code that’s publicly available and anyone can modify it

Angular is an open source, client-side framework that extends HTML using JavaScript. It uses the MVC (model-view-controller) approach to design. Angular is designed for building single page applications (SPA).

What is Angular?

code or operations that happen on the client-side
(e.g. your browser, not the server that hosts the site)

Angular is an open source, client-side framework that extends HTML using JavaScript. It uses the MVC (model-view-controller) approach to design. Angular is designed for building single page applications (SPA).

What is Angular?

reusable code to solve common problems; Mad Libs

Angular is an open source, client-side framework that extends HTML using JavaScript. It uses the MVC (model-view-controller) approach to design. Angular is designed for building single page applications (SPA).

What is Angular?

HyperText Markup Language; webpage guts

Angular is an open source, client-side framework that extends HTML using JavaScript. It uses the MVC (model-view-controller) approach to design. Angular is designed for building single page applications (SPA).

What is Angular?

programming language that's one of three core technologies of the World Wide Web (along with HTML & CSS)

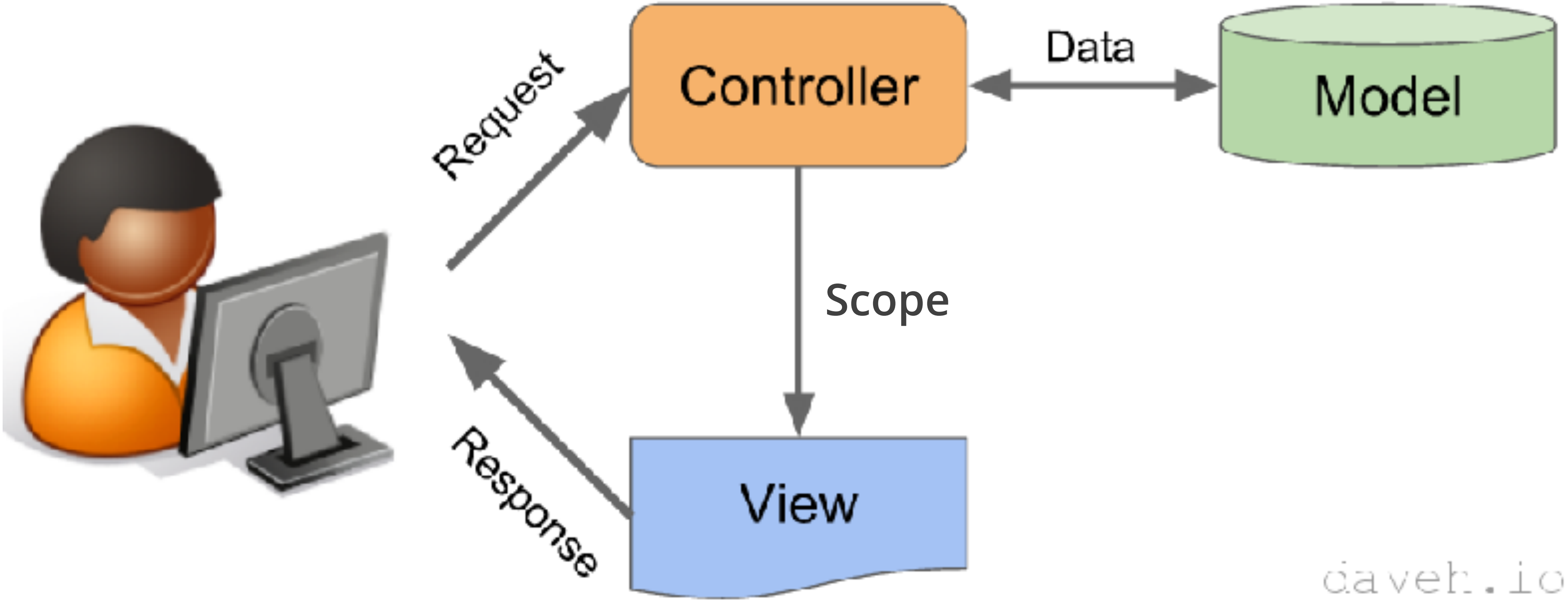
Angular is an open source, client-side framework that extends HTML using JavaScript. It uses the MVC (model-view-controller) approach to design. Angular is designed for building single page applications (SPA).

What is Angular?

software design pattern that encourages separation of concerns

Angular is an open source, client-side framework that extends HTML using JavaScript. It uses the MVC (model-view-controller) approach to design. Angular is designed for building single page applications (SPA).

Model-View-Controller



daveh.io

What is Angular?

Angular is an open source, client-side framework that extends HTML using JavaScript. It uses the MVC (model-view-controller) approach to design. Angular is designed for building single page applications (SPA).

single web page that dynamically updates content based on user interaction with it

Deep Breath!

Deep Breath!

I heard you want to be a web developer
Here are a few devices to test your site

Pros

Cons

Pros

Cons

- Lighter weight than most frameworks

Pros

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps

Cons

Pros

Cons

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding

Pros

Cons

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection

Pros

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection

software design pattern where components are given their dependencies

Cons

Pros

Cons

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection

Pros

Cons

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection
- Easy testing (Jasmine)

Pros

Cons

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection
- Easy testing (Jasmine)

development framework used for testing

Pros

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection
- Easy testing (Jasmine)

Cons

Pros

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection
- Easy testing (Jasmine)

Cons

- Angular 1 & 2 are totally different and not compatible*

Pros

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection
- Easy testing (Jasmine)

Cons

- Angular 1 & 2 are totally different and not compatible*
- Heavy on the jargon

Pros

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection
- Easy testing (Jasmine)

Cons

- Angular 1 & 2 are totally different and not compatible*
- Heavy on the jargon
- Steep learning curve**

Pros

- Lighter weight than most frameworks
- Extends HTML to allow for single page apps
- Two-way data binding
- Dependency injection
- Easy testing (Jasmine)

Cons

- Angular 1 & 2 are totally different and not compatible*
- Heavy on the jargon
- Steep learning curve**
- Not enough documentation***

Quick Start

Quick Start

Directives

Quick Start

Directives

Modules

Quick Start

Directives

Modules

Models

Quick Start

Directives

Modules

Models

Views

Quick Start

Directives

Modules

Models

Views

Controllers

Quick Start

Directives

Modules

Models

Views

Controllers

Data binding

Quick Start

Directives

Modules

Models

Views

Controllers

Data binding

Set-up

Directives

Directives

Attributes that extend HTML

Directives

Attributes that extend HTML

<tag attribute=“value”>content</tag>

Directives

Attributes that extend HTML

Directives

Attributes that extend HTML

A function that executes when the compiler encounters it in the DOM

Directives

Attributes that extend HTML

A function that executes when the compiler encounters it in the DOM

compiler teaches browser what to do with directives

Directives

Attributes that extend HTML

A function that executes when the compiler encounters it in the DOM

created when page is loaded, the DOM allows JavaScript to change HTML/CSS on the page

compiler teaches browser what to do with directives

Directives

Attributes that extend HTML

A function that executes when the compiler encounters it in the DOM

created when page is loaded, the DOM allows JavaScript to change HTML/CSS on the page

Directives

Attributes that extend HTML

A function that executes when the compiler encounters it in the DOM

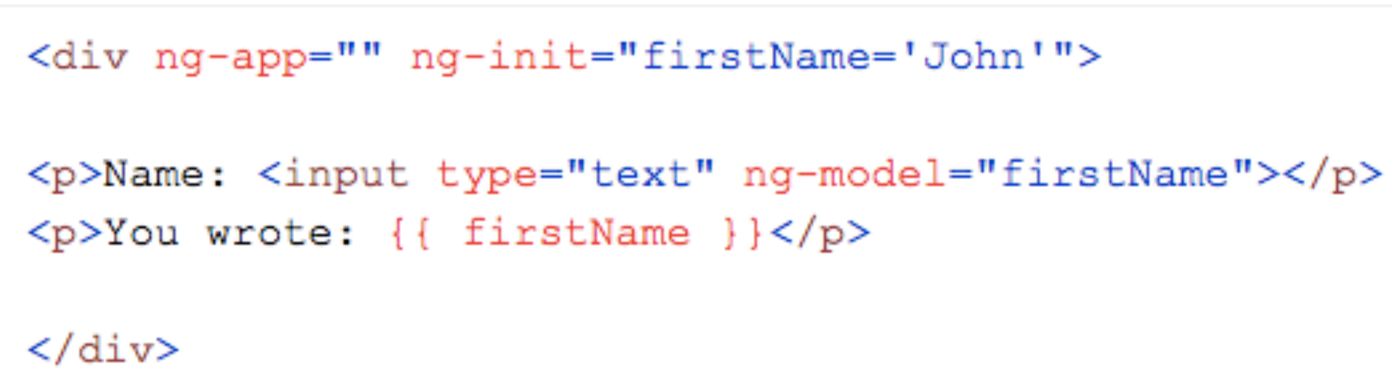
Directives

Attributes that extend HTML

A function that executes when the compiler encounters it in the DOM

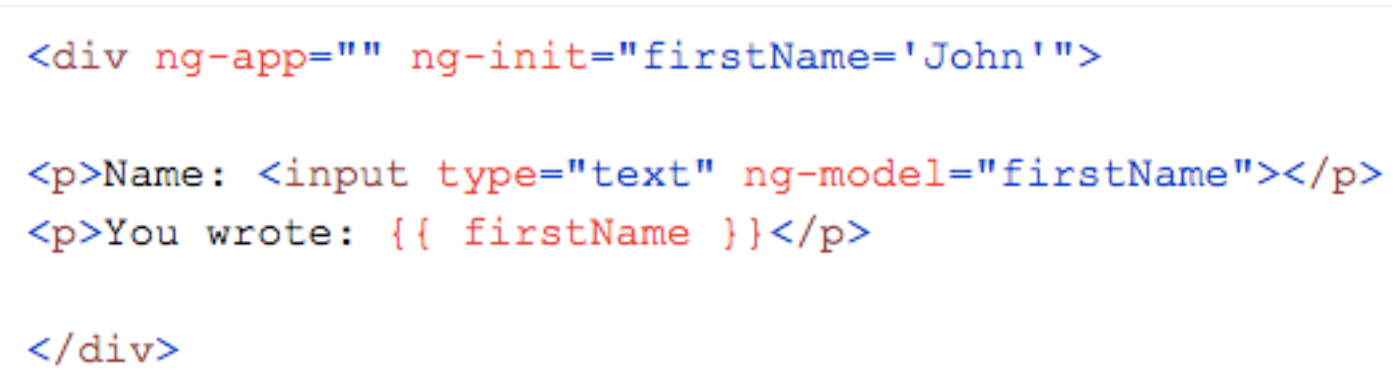
Angular comes with built-in directives, but you can also build custom directives

Examples of Directives



```
<div ng-app="" ng-init="firstName='John'">

<p>Name: <input type="text" ng-model="firstName"></p>
<p>You wrote: {{ firstName }}</p>

</div>
```

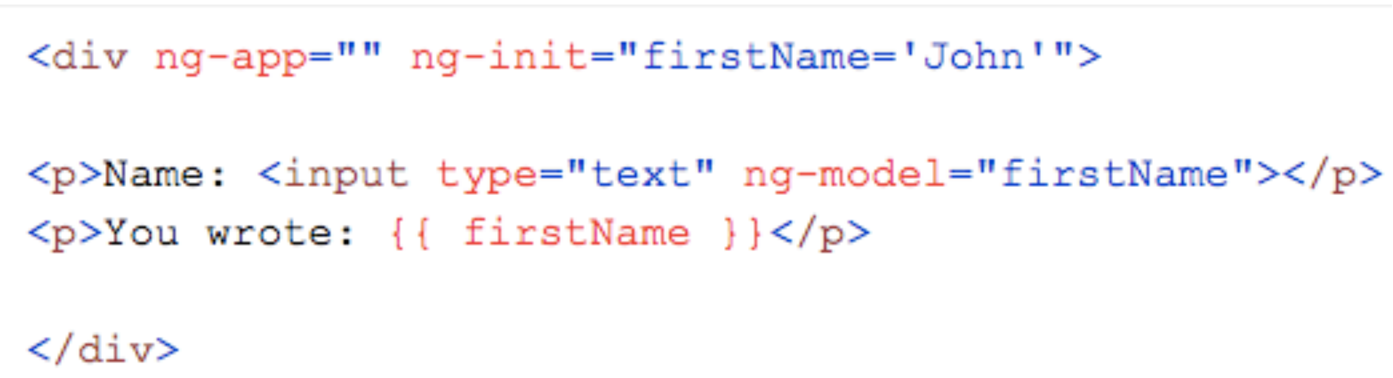
Examples of Directives



```
<div ng-app="" ng-init="firstName='John'">

<p>Name: <input type="text" ng-model="firstName"></p>
<p>You wrote: {{ firstName }}</p>

</div>
```

ngApp, ngInit, and ngModel are all Angular directives

Examples of Directives

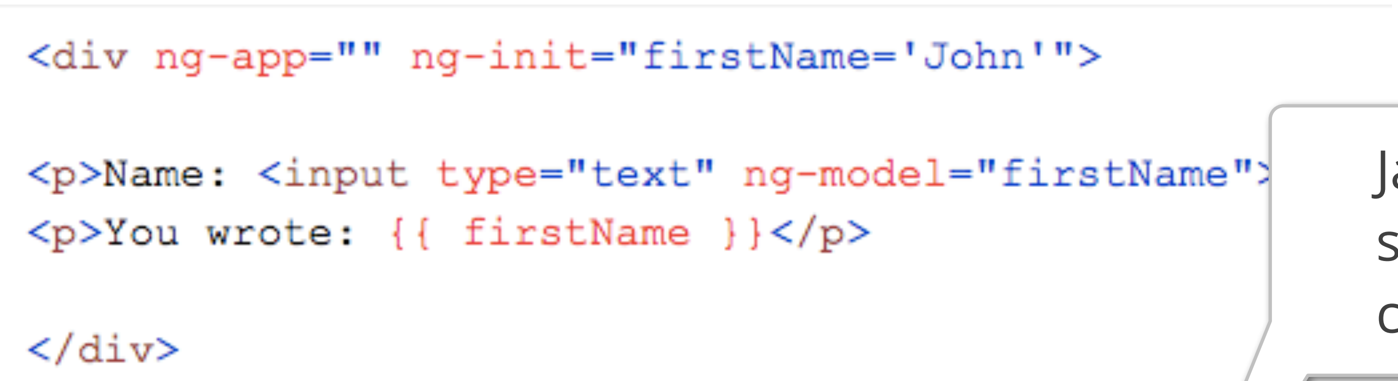


```
<div ng-app="" ng-init="firstName='John'">

<p>Name: <input type="text" ng-model="firstName"></p>
<p>You wrote: {{ firstName }}</p>

</div>
```

ngApp, ngInit, and ngModel are all Angular directives

(and anything in {{ }} is an expression)

Examples of Directives



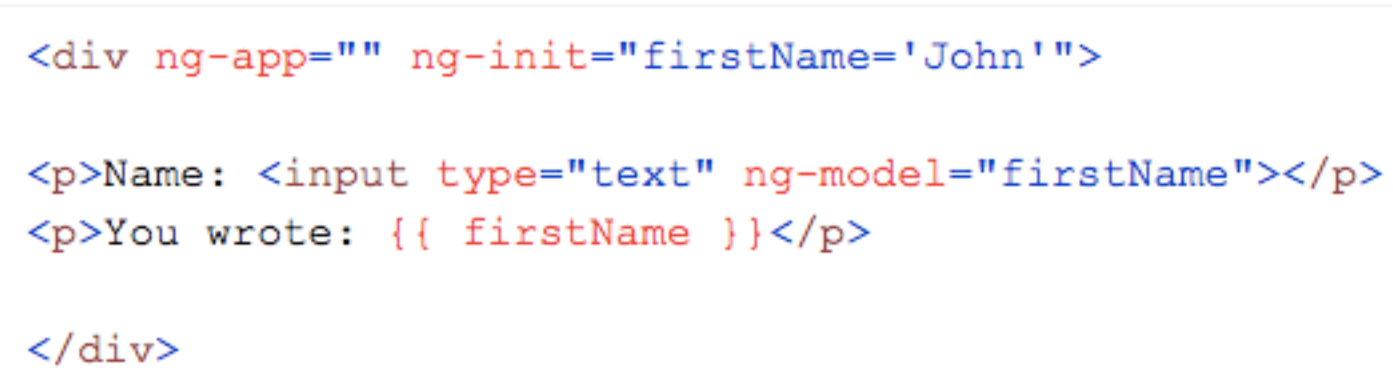
```
<div ng-app="" ng-init="firstName='John'">

<p>Name: <input type="text" ng-model="firstName">
<p>You wrote: {{ firstName }}</p>

</div>
```

JavaScript-like code snippet that's in curly brackets

ngApp, ngInit, and ngModel are all Angular directives

(and anything in {{ }} is an expression)

Examples of Directives



```
<div ng-app="" ng-init="firstName='John'">

<p>Name: <input type="text" ng-model="firstName"></p>
<p>You wrote: {{ firstName }}</p>

</div>
```

ngApp, ngInit, and ngModel are all Angular directives

(and anything in {{ }} is an expression)

Modules

Modules

“An app within an app”

Modules

“An app within an app”

Separation of concerns

Modules

“An app within an app”

Separation of concerns

how pieces of the app are separated into reusable chunks

Modules

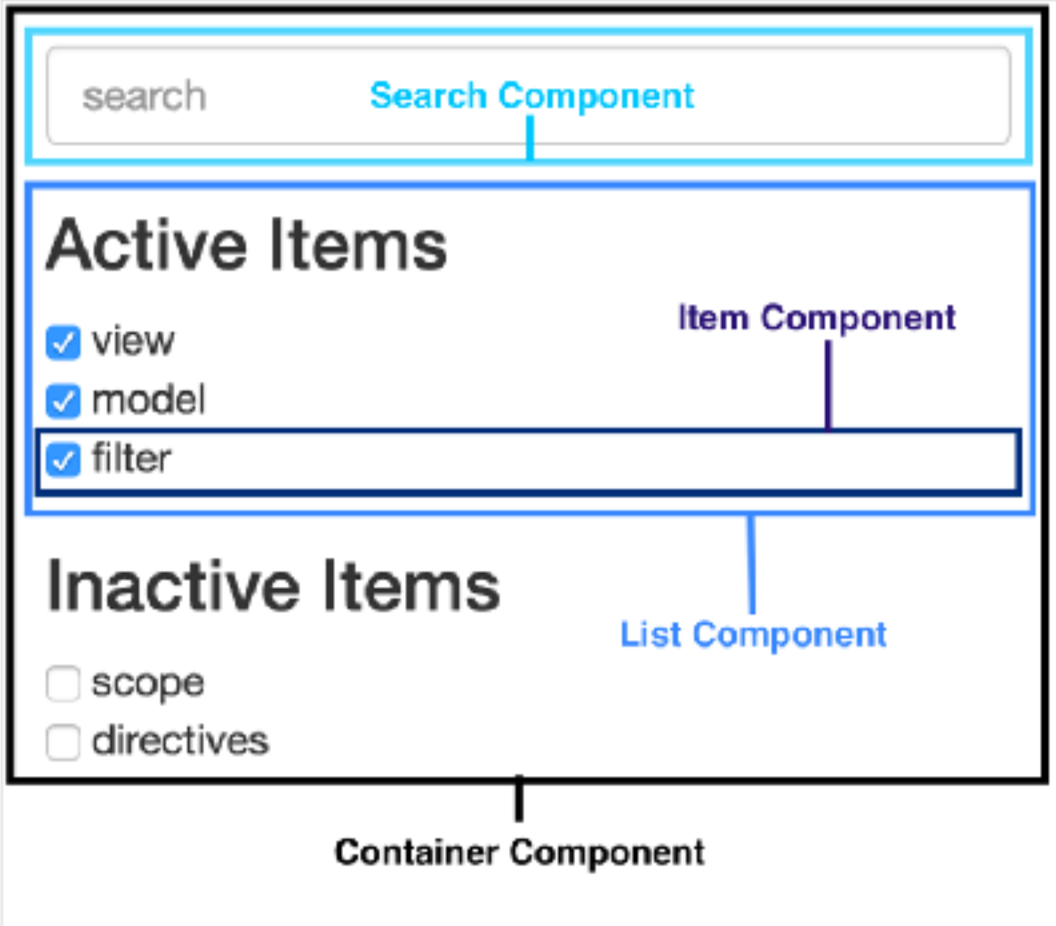
“An app within an app”

Separation of concerns

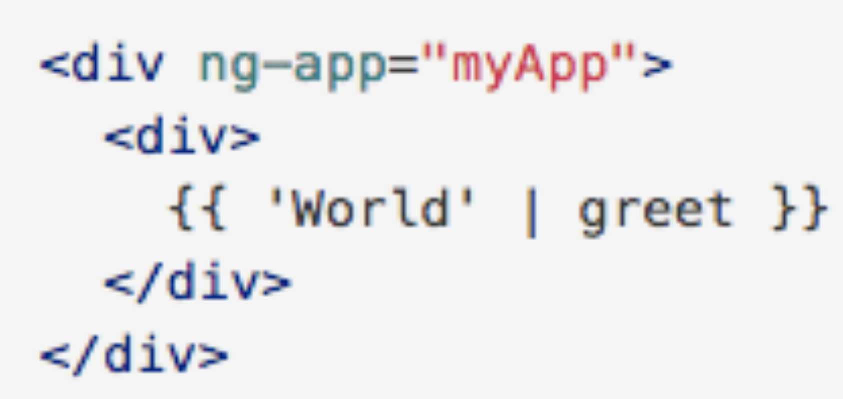
Modules

“An app within an app”

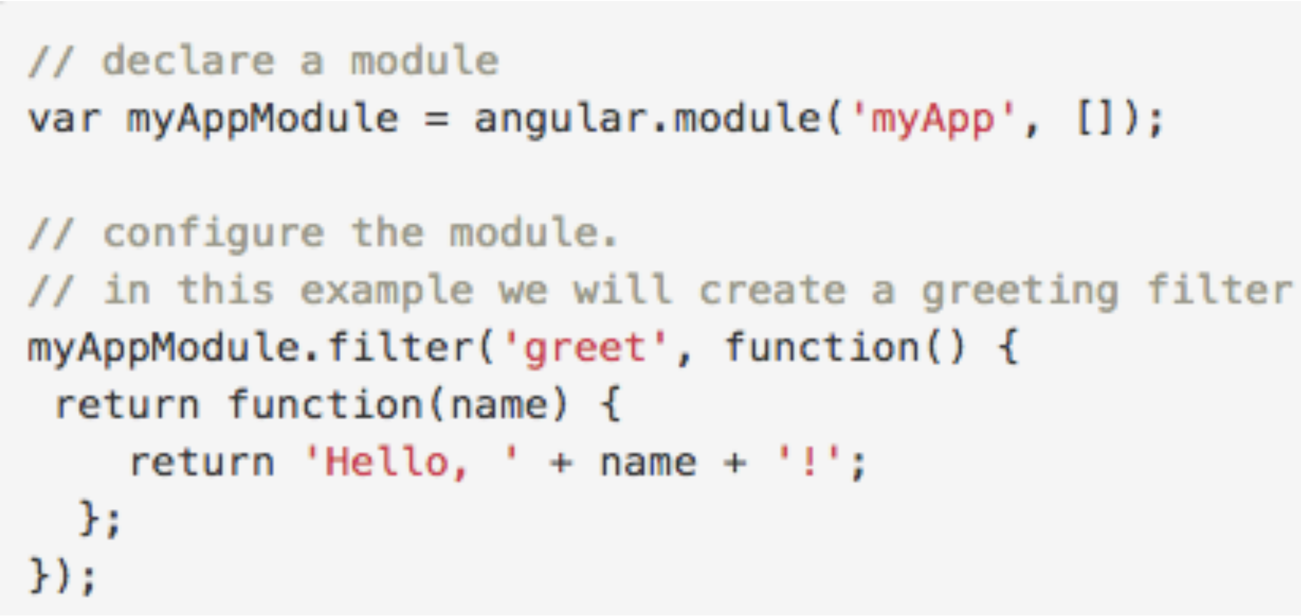
Separation of concerns



Example Module

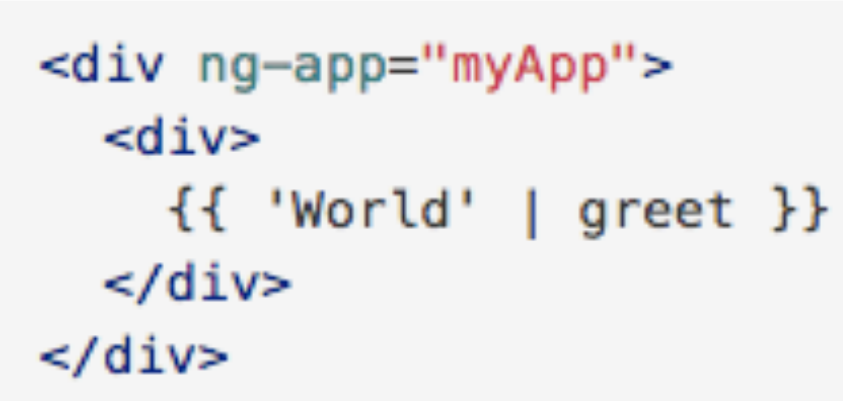


```
<div ng-app="myApp">
  <div>
    {{ 'World' | greet }}
  </div>
</div>
```

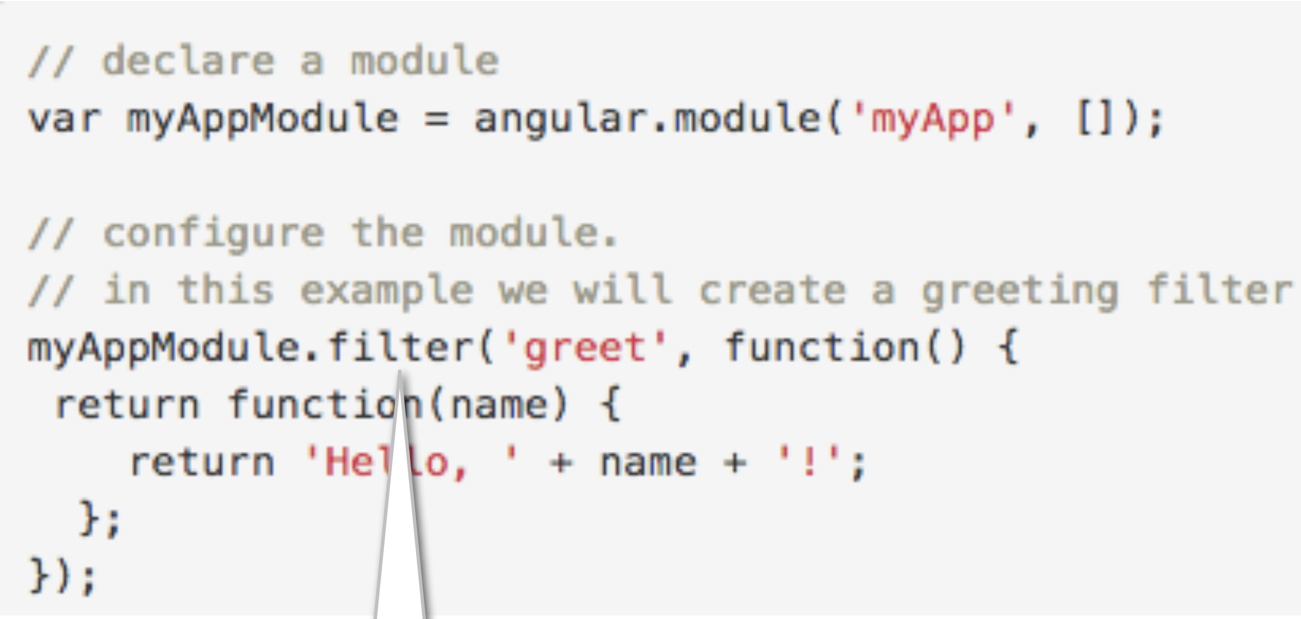


```
// declare a module
var myAppModule = angular.module('myApp', []);

// configure the module.
// in this example we will create a greeting filter
myAppModule.filter('greet', function() {
 return function(name) {
    return 'Hello, ' + name + '!';
  };
});
```

Example Module



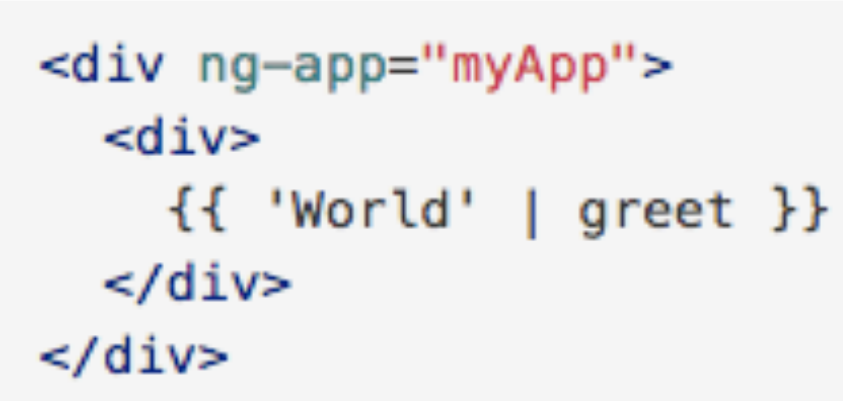
```html
<div ng-app="myApp">
  <div>
    {{ 'World' | greet }}
  </div>
</div>
```



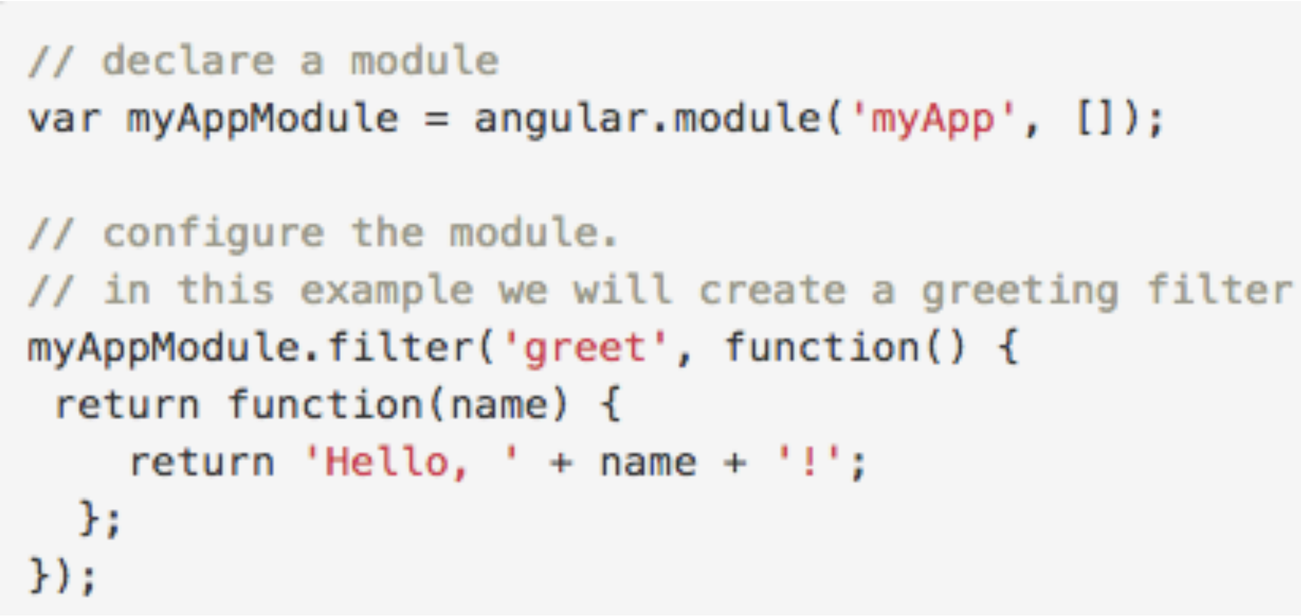
```javascript
// declare a module
var myAppModule = angular.module('myApp', []);

// configure the module.
// in this example we will create a greeting filter
myAppModule.filter('greet', function() {
 return function(name) {
    return 'Hello, ' + name + '!';
  };
});
```

formats or transforms data

Example Module



```html
<div ng-app="myApp">
  <div>
    {{ 'World' | greet }}
  </div>
</div>
```



```javascript
// declare a module
var myAppModule = angular.module('myApp', []);

// configure the module.
// in this example we will create a greeting filter
myAppModule.filter('greet', function() {
 return function(name) {
  return 'Hello, ' + name + '!';
 };
});
```

Let’s Build Stuff!

Let's Build Stuff!

Static module

Let's Build Stuff!

Static module

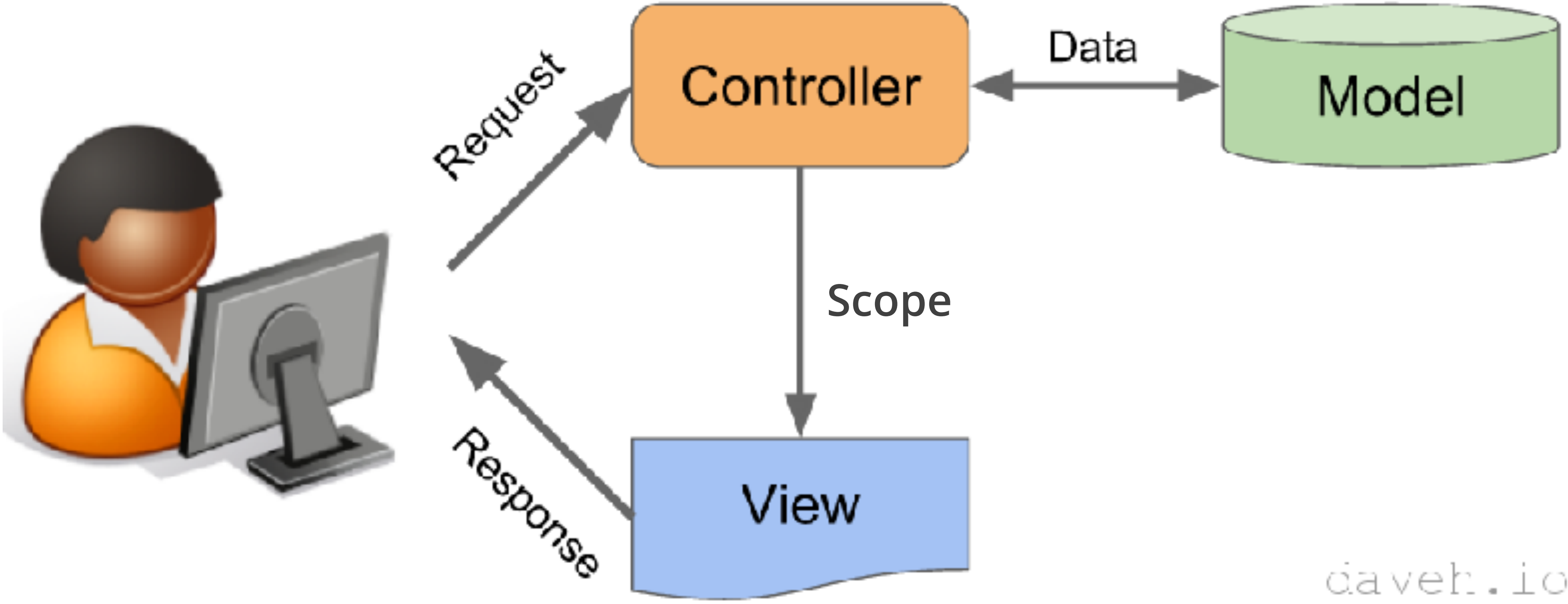
Let's Build Stuff!

Static module → Dynamic module

Deep Breath!

Deep Breath!

Model-View-Controller



Controller

Model

View

Data

Request

Response

Scope

daveh.io

Model

Model

Collection of data available for the application

Model

Collection of data available for the application

ngModel directive binds value of HTML controls/elements (view) to app data (model)

Model

Collection of data available for the application

ngModel directive binds value of HTML controls/elements (view) to app data (model)

Can also provide validation of data

Model

Collection of data available for the application

ngModel directive binds value of HTML controls/elements (view)

checking for correctness and meaningfulness of the data a user inputs

Can also provide validation of data

Model

Collection of data available for the application

ngModel directive binds value of HTML controls/elements (view) to app data (model)

Can also provide validation of data

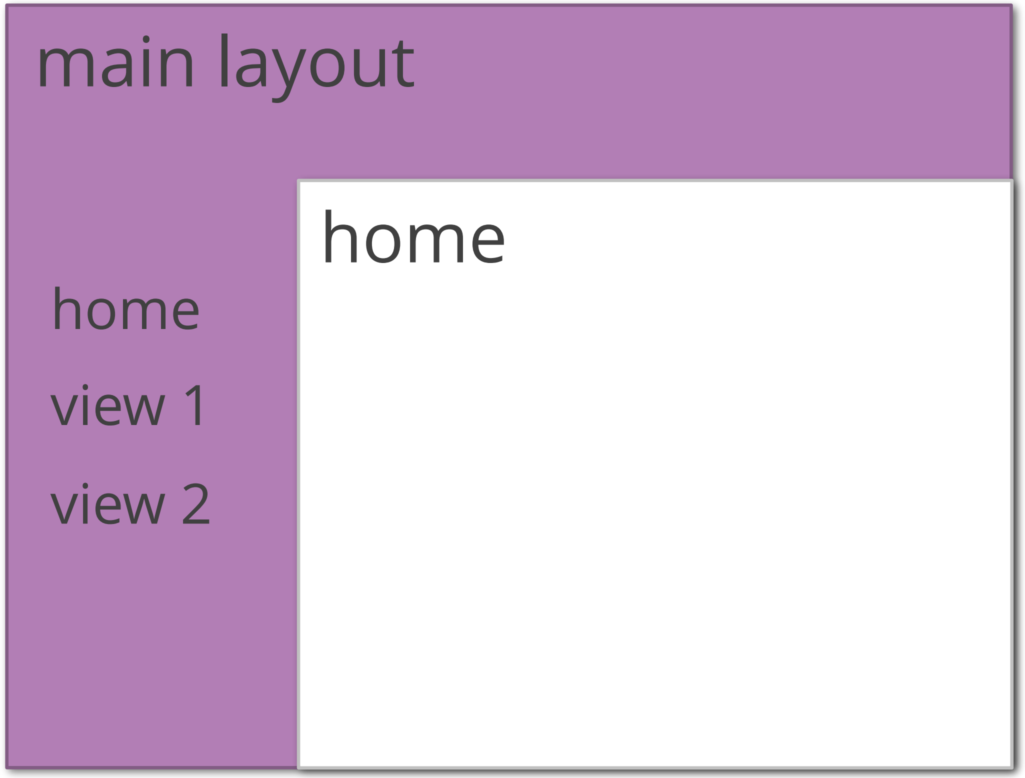
View

View

Single-page application: can load different pages (*views*) without reloading the entire page

View

Single-page application: can load different pages (*views*) without reloading the entire page



main layout

home

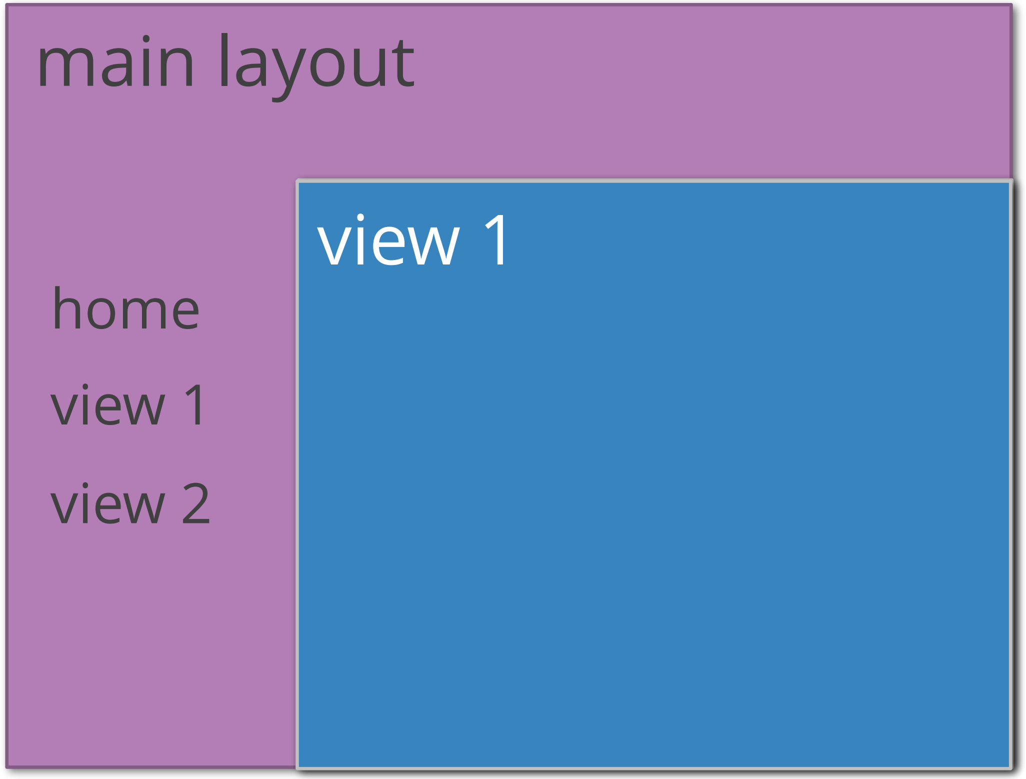
view 1

view 2

home

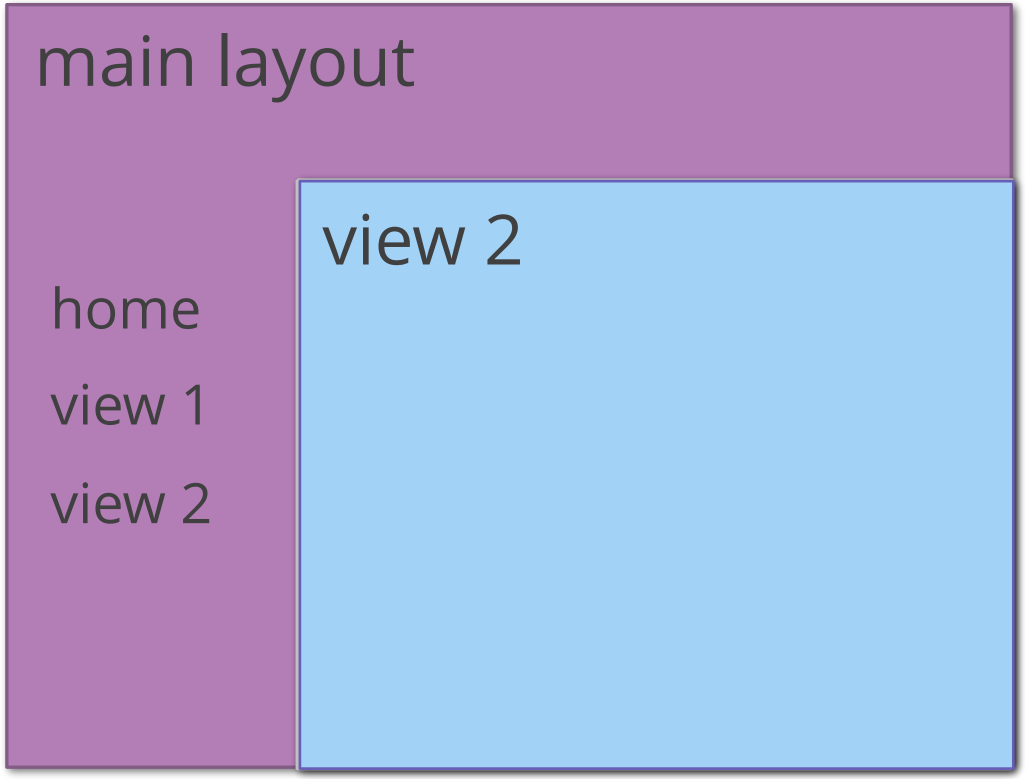
View

Single-page application: can load different pages (*views*) without reloading the entire page



View

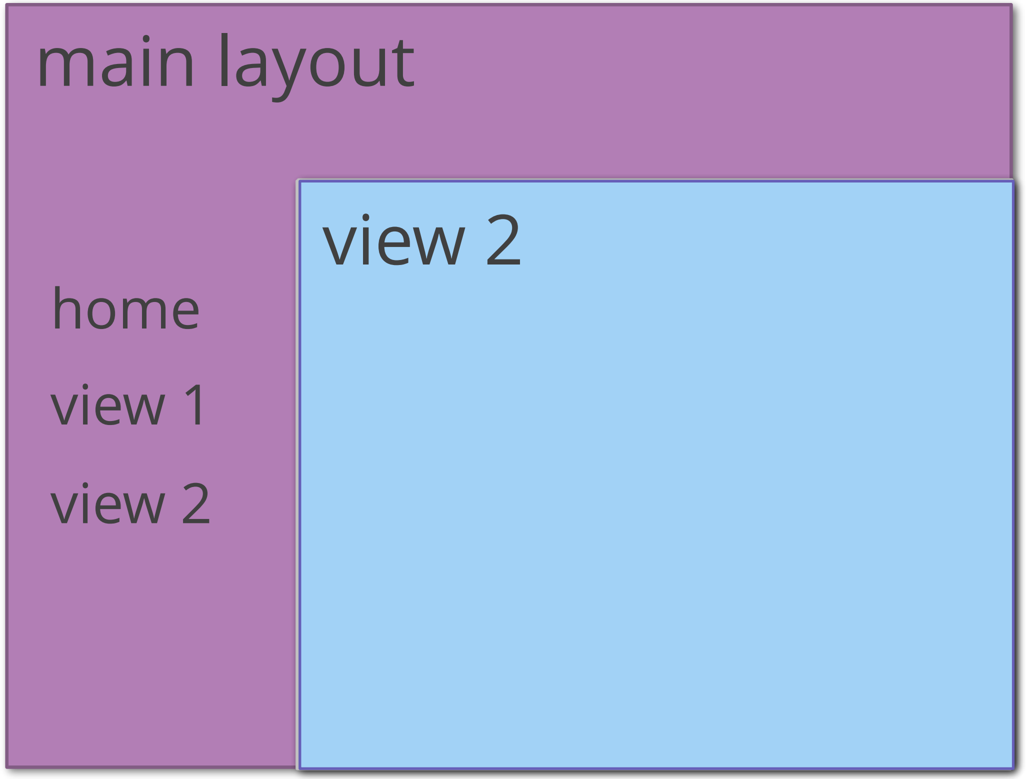
Single-page application: can load different pages (*views*) without reloading the entire page



View

Single-page application: can load different pages (*views*) without reloading the entire page

This is accomplished through the ngRoute module (which includes ngView directive)



main layout

home

view 1

view 2

view 2

Controller

Controller

Defines the behavior behind DOM elements

Controller

Defines the behavior behind DOM elements

Controls interactions between model and view

Controller

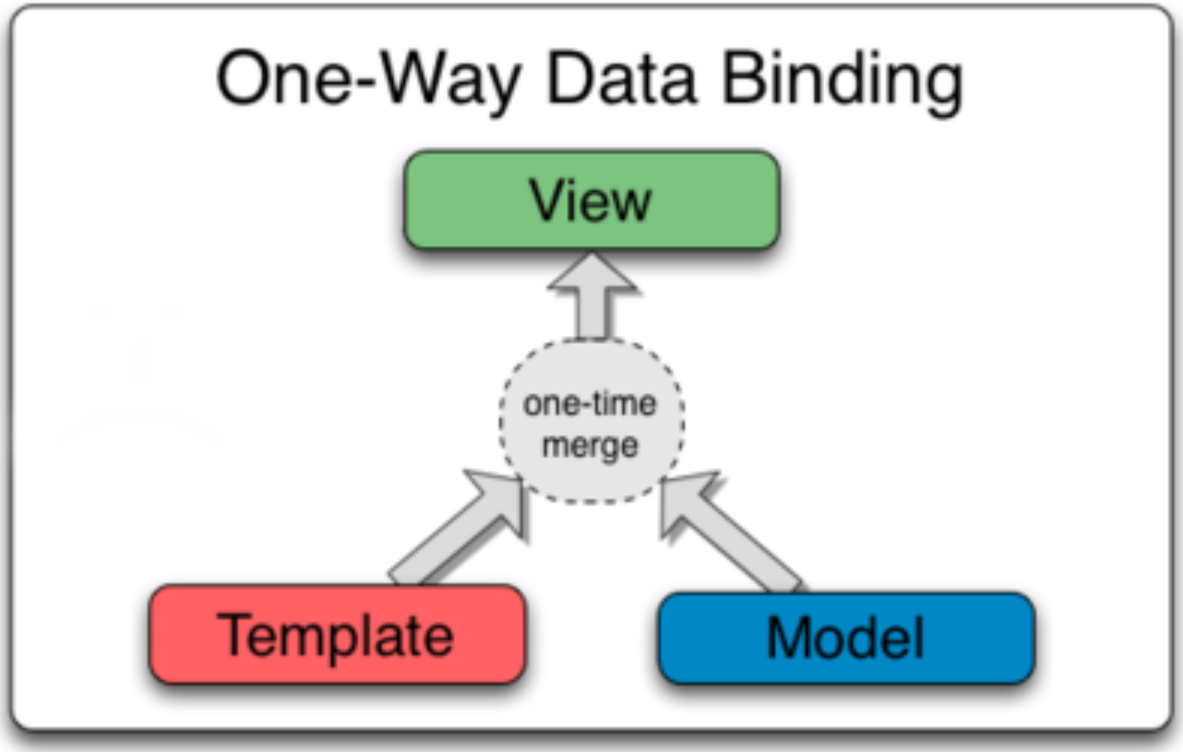
Defines the behavior behind DOM elements

Controls interactions between model and view

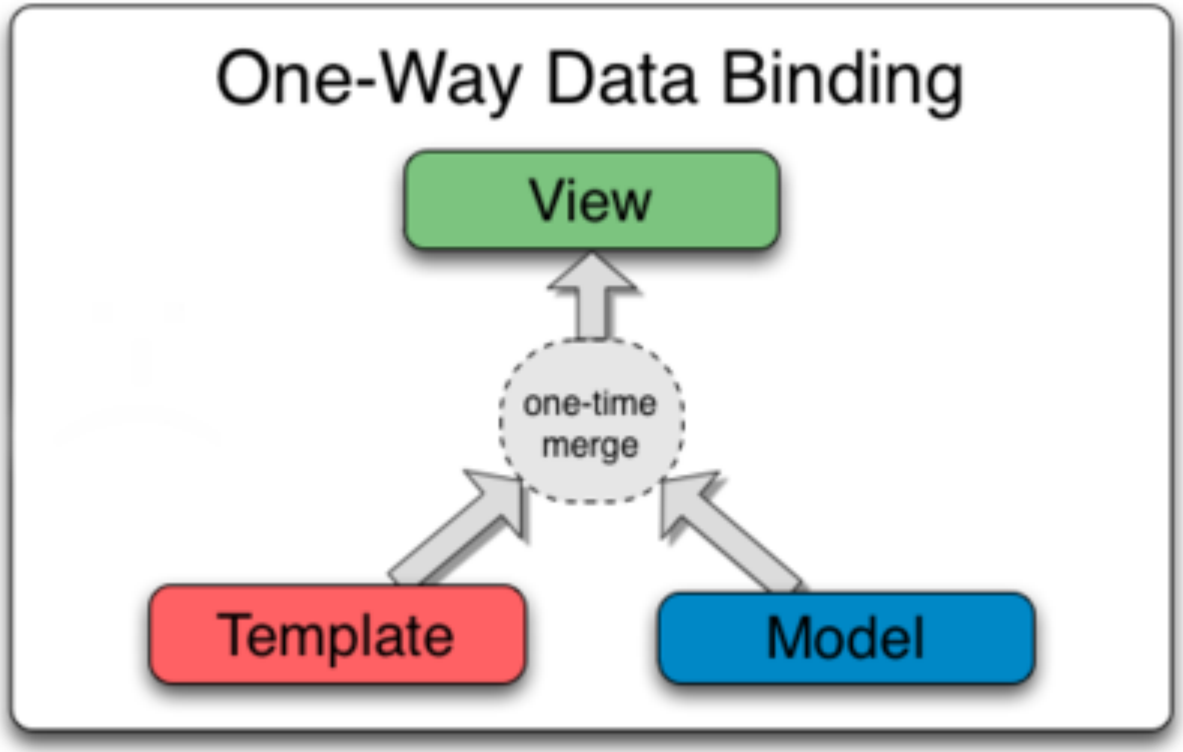
Separate controllers for separate views

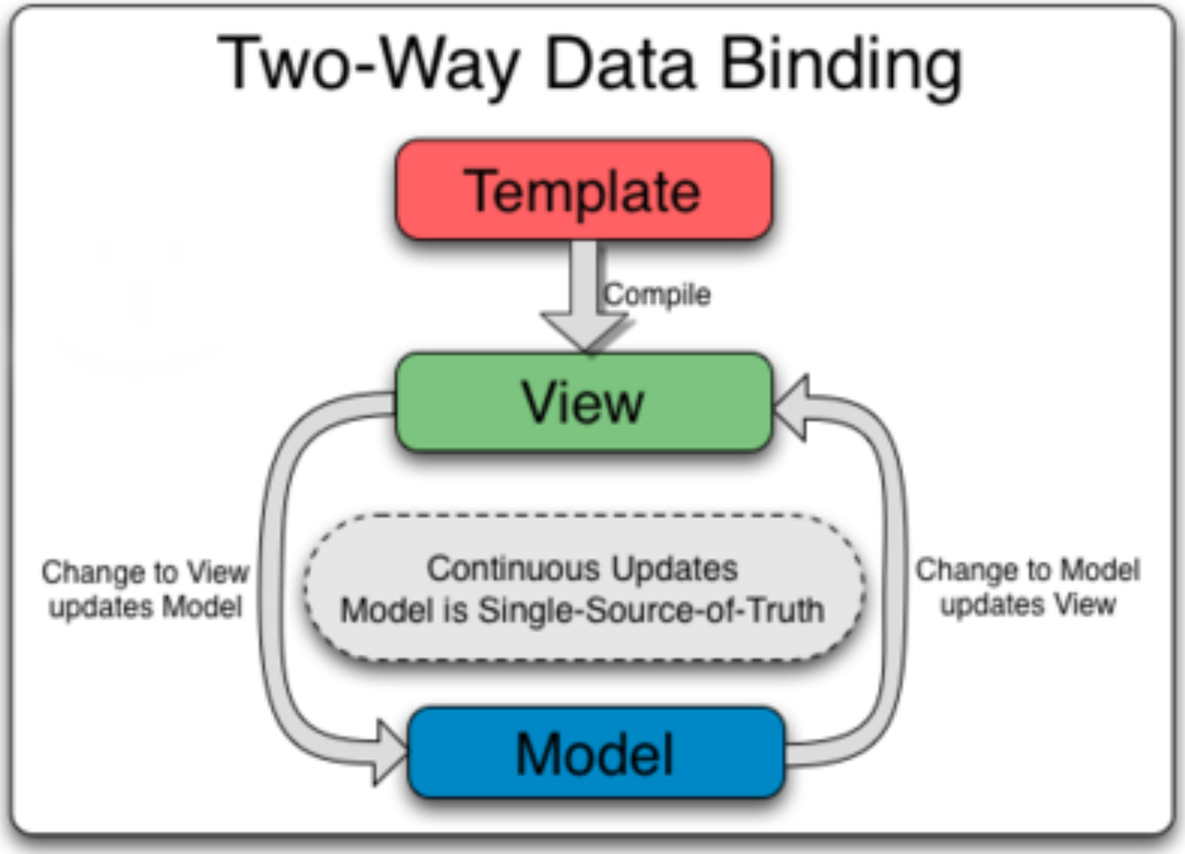
Data Binding

Data Binding



Data Binding




Two-Way Data Binding
Template
Compile
View
Change to View updates Model
Continuous Updates
Model is Single-Source-of-Truth
Change to Model updates View
Model

Set Up

Set Up

Add Angular to page via CDN

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"></script>
```

Set Up

content delivery network

Add Angular to page via CDN

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"></script>
```

Set Up

Add Angular to page via CDN

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"></script>
```

Set Up

Add Angular to page via CDN

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"></script>
```

Set Up

Add Angular to page via CDN

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"></script>
```

Set Up

Add Angular to page via CDN

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"></script>
```

That's it!

Style Guide

Style Guide

Rules or guidelines on how to structure and write your code, helps people understand your code because it’s standardized

Style Guide

Rules or guidelines on how to structure and write your code, helps people understand your code because it’s standardized

Two options for Angular style guides:

Style Guide

Rules or guidelines on how to structure and write your code, helps people understand your code because it's standardized

Two options for Angular style guides:

1. Todd Motto (https://github.com/toddmotto/angular-styleguide)

Style Guide

Rules or guidelines on how to structure and write your code, helps people understand your code because it’s standardized

Two options for Angular style guides:

1. Todd Motto (https://github.com/toddmotto/angular-styleguide)
2. John Papa (https://github.com/johnpapa/angular-styleguide)

Let's Build Some More Stuff!

To-do list

Questions?

Contact

Resources

Contact

Resources

geekscat.com

https://github.com/kaymckelly

kayasmckelly@gmail.com

Contact

geekscat.com

https://github.com/kaymckelly

kayasmckelly@gmail.com

Resources

- angularjs.org/
- github.com/angular/angular.js
- codecademy.com/learn/learn-angularjs
- github.com/johnpapa/angular-styleguide

Contact

geekscat.com

https://github.com/kaymckelly

kayasmckelly@gmail.com

Resources

- angularjs.org/
- github.com/angular/angular.js
- codecademy.com/learn/learn-angularjs
- github.com/johnpapa/angular-styleguide

Thanks for coming!