HACK YALE

< ADVANCED JAVASCRIPT />

WWW.HACKYALE.COM

HACK YALE

< ADVANCED JAVASCRIPT />

DAY 3

ONWARD TO ANGULAR

IT'S ALMOST OVER :'(

The agenda:

- Homework review
- Install Python
- Dependency injection
- The digest cycle
- Routes and app structure
- Lez make a game

HOMEWORK REVIEW

HACK YALE

COMPARE NOTES

CHECK IN WITH THE PEOPLE NEAR YOU —
WHAT DID YOU DO SIMILARLY?
WHAT WAS DIFFERENT?

INSTALL PYTHON

OPEN TERMINAL/CMD, TYPE 'PYTHON'
(CTRL-D TO QUIT)

IF THERE'S AN ERROR:
PYTHON.ORG/DOWNLOADS/

MORE ABOUT ANGULAR

HACK YALE

DEPENDENCY INJECTION

MODULAR DESIGN

DEPENDENCY INJECTION

Like we discussed last time, Angular is powerful because it allows you to write modular code

- Create *components* that depend on other components
  - For example, `TodoCtrl` depends on `$scope`
- Specify which components you depend on

DI: SYNTAX

```
1 angular.module("Todo", [])
2     .controller("TodoCtrl", function($scope) {
```

Easy!

DI: UNDER THE HOOD

When you inject something, Angular behind the scenes checks for all injectable things (called *services*)

- Name matters! If you depend on $scope, that's different from scope or $Scope
- Note that though you're passing a parameter into a function, that parameter has *already been defined*
- All Angular-defined services start with a $


THE DIGEST CYCLE
$SCOPE.APPLY

REMEMBER DATA BINDING?

How does it *automagically* work?

- Behind the scenes, Angular sets up *watchers* for your variables
  - Watchers are functions that run when a variable value changes
- Every time there's an Angular event, Angular checks if any of the variables have changed
  - If so, it calls the watcher functions

WATCHERS

```
$scope.$watch('myVar', function(newV, oldV) {
  console.log('myVar changed from', newV, "to", oldV);
});
```

This function will run every time the value of myVar changes

$SCOPE.DIGEST()

- The $digest function on the $scope service checks all the watched variables to see if they've changed
- This only happens after Angular events
  - Angular events: ng-click, ng-mouseover, etc…

WORKING OUTSIDE ANGULAR

- When you call non-Angular methods, sometimes the $digest function isn't called
  - Most common examples: setTimeout, jQuery event handlers
- To fix that, you have run the $digest cycle yourself. This is done by using $scope.$apply()
  - $scope.$apply() just calls $rootScope.$digest()

WILL THIS WORK?

```
20  $scope.message = "I'm here now";
21  setTimeout(function() {
22    $scope.message = "I came later";
23  }, 2000)
```

What will $scope.message be after 3 seconds?

THE FIX

```
20  $scope.message = "I'm here now";
21  setTimeout(function() {
22      $scope.message = "I came later";
23      $scope.$apply();
24  }, 2000)
```

WELL THAT'S ANNOYING...

- Good news is, Angular comes with built-in services that do this for you
  - $scope.$on('click'), $timeout, etc.

THE FIX

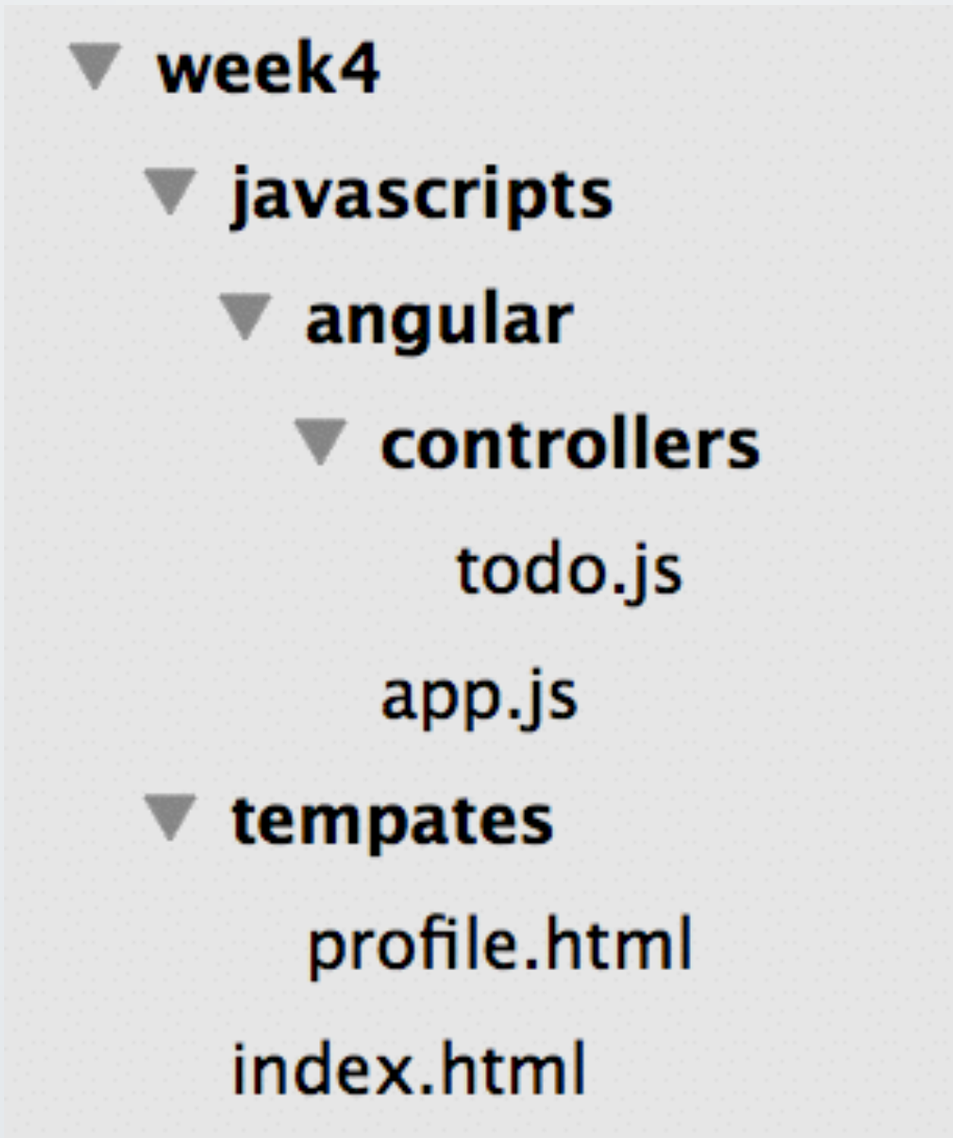
```
20  ...controller("MsgCtrl", function($scope, $timeout) {
21    $scope.message = "I'm here now";
22    $timeout(function() {
23      $scope.message = "I came later";
24    }, 2000)
25  })
```

APP STRUCTURE

FOLDERS GALORE

GROWING YOUR APP

- As your application grows, you'll want to factor your code out into separate files and folders.



ORDER MATTERS...KINDA

- It's important to include whichever files contains the line angular.module('name', [dependencies…]) first, then the order doesn't matter
- That file is usually app.js (or named your_app_name.js)

ORDER MATTERS...KINDA

```
1  <!DOCTYPE html>
2  <head>
3    <title>My app</title>
4    <!-- Include Angular -->
5    <script src="http://ajax.googleapis.com/ajax/libs/angu
       min.js"></script>
6    <script src="./angular/my_app.js"></script>
7    <script src="./angular/controllers/todo.js"></script>
8  </head>
9  <body ng-app="MyApp">
10
11 </body>
12 </html>
```

ORDER MATTERS...KINDA

```
1  // angular/my_app.js
2  angular.module("MyApp", [])
3      .config(function() {
4          // App configuration
5  }).run(function() {
6          // Initialization code
7          // to be run once when
8          // the app starts up.
9  });
```

ORDER MATTERS...KINDA

```
1  // angular/controllers/todo.js
2  angular.module("MyApp") // Note: no dependencies
3    .controller("TodoCtrl", function($scope) {
4      // Code for controller here
5    })
```


UI ROUTER
FRONT-END ROUTES IN A JIFFY

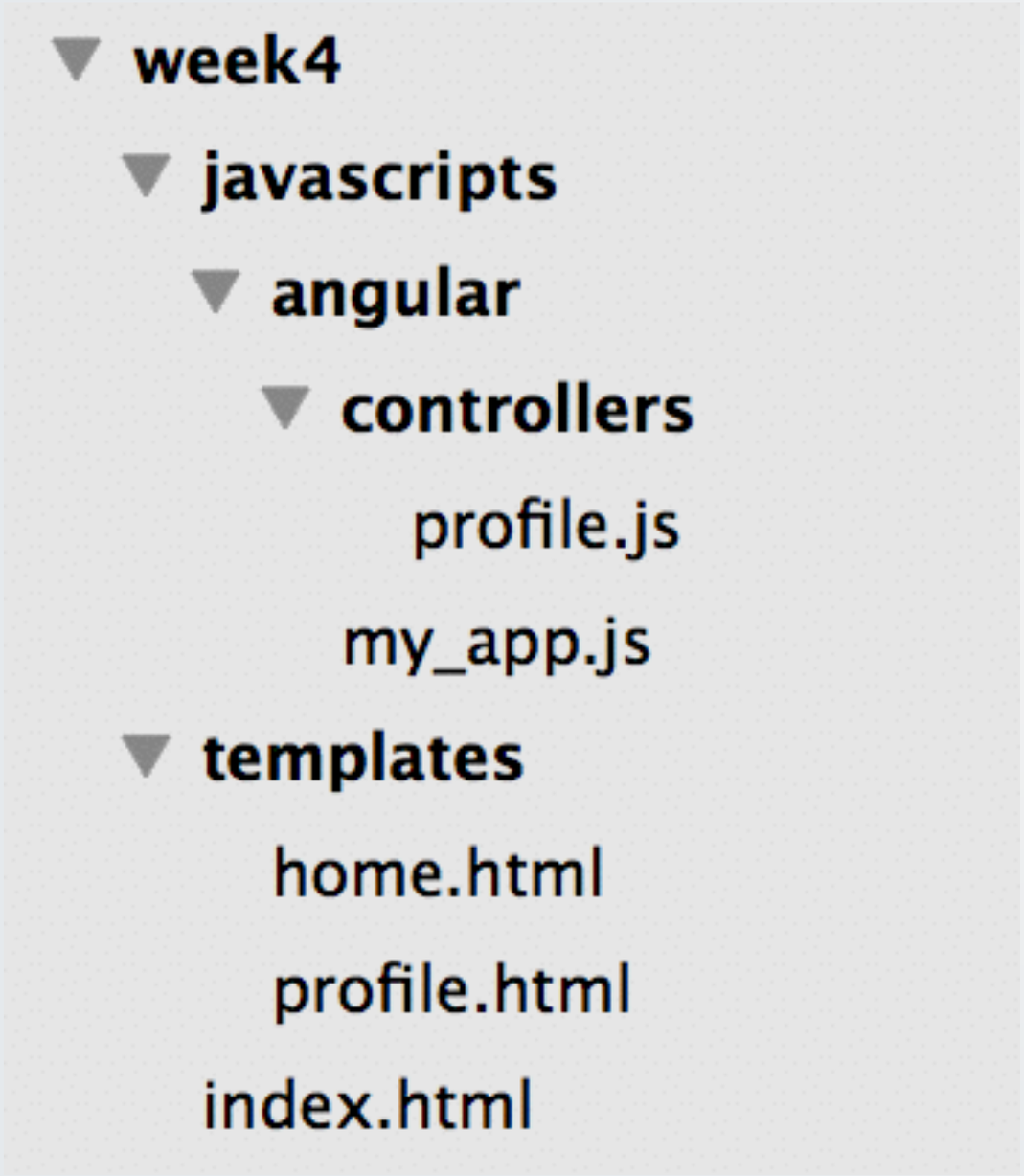
UI ROUTER

- The router that comes with Angular kind of sucks
- But no worries, the "community" has built one that's better!
  - One of the great things about Angular

UI ROUTER

- Include it via CDN
- Define a set of *states* in angular.module.config
  - Name, URL, template view, controller
    - Like a roadmap of your entire application
- Make the templates and controllers
- Link with <a ui-sref="state-name">
- Everything just works!

EXAMPLE: APP STRUCTURE



index.html: the root of your page—include all javascript, make a <ui-view>

angular: the folder for all your angular code

templates: where your HTML really goes

EXAMPLE: INDEX.HTML

Include app.js first

```
1  <!DOCTYPE html>
2  <head>
3    <title>My app</title>
4    <!-- Include Angular -->
5    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.25/an
       min.js"></script>
6    <!-- Include UI Router -->
7    <script src="http://cdnjs.cloudflare.com/ajax/libs/angular-ui-router/
       11/angular-ui-router.min.js"></script>
8    <script src="./javascripts/angular/my_app.js"></script>
9    <script src="./javascripts/angular/controllers/profile.js"></script>
10 </head>
11 <body ng-app="MyApp">
12   <div class="ui-view"></div>
13 </body>
14 </html>
```

EXAMPLE: INDEX.HTML

```
1  <!DOCTYPE html>
2  <head>
3    <title>My app</title>
4    <!-- Include Angular -->
5    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.25/an
       min.js"></script>
6    <!-- Include UI Router -->
7    <script src="http://cdnjs.cloudflare.com/ajax/libs/angular-ui-router/
       11/angular-ui-router.min.js"></script>
8    <script src="./javascripts/angular/my_app.js"></script>
9    <script src="./javascripts/angular/controllers/profile.js"></script>
10 </head>
11 <body ng-app="MyApp">
12   <div class="ui-view"></div>
13 </body>
14 </html>
```

Then each of your other files individually

EXAMPLE: INDEX.HTML

```
1  <!DOCTYPE html>
2  <head>
3    <title>My app</title>
4    <!-- Include Angular -->
5    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.25/an
       min.js"></script>
6    <!-- Include UI Router -->
7    <script src="http://cdnjs.cloudflare.com/ajax/libs/angular-ui-router/
       11/angular-ui-router.min.js"></script>
8    <script src="./javascripts/angular/my_app.js"></script>
9    <script src="./javascripts/angular/controllers/profile.js"></script>
10 </head>
11 <body ng-app="MyApp">
12   <div class="ui-view"></div>
13 </body>
14 </html>
```

The ui-view is Angular's gateway into your HTML

EXAMPLE: APP.JS

```
1  // angular/my_app.js
2  angular.module("MyApp", ["ui.router"])
3    .config(function($stateProvider, $urlRouterProvider) {
4      $stateProvider
5        .state("home", {
6          url: "/",
7          templateUrl: "./templates/home.html"
8        })
9        .state("profile", {
10         url: "/profile",
11         templateUrl: "./templates/profile.html",
12         controller: "ProfileCtrl"
13       });
14     // Make the default route "/" instead of nothing
15     $urlRouterProvider.otherwise("/");
16   });
```

Depend on UI Router

EXAMPLE: APP.JS

```
1  // angular/my_app.js
2  angular.module("MyApp", ["ui.router"])
3    .config(function($stateProvider, $urlRouterProvider) {
4      $stateProvider
5        .state("home", {
6          url: "/",
7          templateUrl: "./templates/home.html"
8        })
9        .state("profile", {
10         url: "/profile",
11         templateUrl: "./templates/profile.html",
12         controller: "ProfileCtrl"
13       });
14     // Make the default route "/" instead of nothing
15     $urlRouterProvider.otherwise("/");
16   });
```

Inject in $stateProvider

EXAMPLE: APP.JS

```
1  // angular/my_app.js
2  angular.module("MyApp", ["ui.router"])
3    .config(function($stateProvider, $urlRouterProvider) {
4      $stateProvider
5        .state("home", {
6          url: "/",
7          templateUrl: "./templates/home.html"
8        })
9        .state("profile", {
10         url: "/profile",
11         templateUrl: "./templates/profile.html",
12         controller: "ProfileCtrl"
13       });
14     // Make the default route "/" instead of nothing
15     $urlRouterProvider.otherwise("/");
16 });
```

Define your states

EXAMPLE: ANGULAR/APP.JS

```
1  // angular/my_app.js
2  angular.module("MyApp", ["ui.router"])
3    .config(function($stateProvider, $urlRouterProvider) {
4      $stateProvider
5        .state("home", {
6          url: "/",
7          templateUrl: "./templates/home.html"
8        })
9        .state("profile", {
10         url: "/profile",
11         templateUrl: "./templates/profile.html",
12         controller: "ProfileCtrl"
13       });
14     // Make the default route "/" instead of nothing
15     $urlRouterProvider.otherwise("/");
16   });
```

View-specific HTML goes in templates

EXAMPLE: APP.JS

```
1  // angular/my_app.js
2  angular.module("MyApp", ["ui.router"])
3    .config(function($stateProvider, $urlRouterProvider) {
4      $stateProvider
5        .state("home", {
6          url: "/",
7          templateUrl: "./templates/home.html"
8        })
9        .state("profile", {
10         url: "/profile",
11         templateUrl: "./templates/profile.html",
12         controller: "ProfileCtrl"
13       });
14     // Make the default route "/" instead of nothing
15     $urlRouterProvider.otherwise("/");
16 });
```

Set a default route

EXAMPLE: TEMPLATES/HOME.HTML

```
1  <h1>Home!</h1>
2  <a ui-sref="profile">To my profile</a>
```

Link with ui-sref and the state name

CODING CHALLENGE!

HACK YALE

CODING CHALLENGE: PART 1

- Using the app structure we just learned, put a blue square on your home page. Using `setInterval`, make it so that the box appears at a random coordinate on the screen every second.
- Is there an Angular version of `setInterval`?

CODING CHALLENGE: PART 2

- Add a timer that shows the seconds elapsed since the page was loaded, using `setInterval` and `$scope.$apply()` (or `$interval`). Show that number on the page.

CODING CHALLENGE: PART 3

- Make it so if you can click the box, you get a point
- When you get 10 points, redirect to the "winner" page!
  - Hint: use $state.go() to go to another page from a controller
  - Can you get the points and time elapsed to show up on the winner page?