Beta Edition

MEAN Full Stack Development

SASS, JavaScript, jQuery and Angular

Daniel Deverell

Ahoy Thar Matey!

Building the Pirate Portfolio view

Pirate Portfolio Page

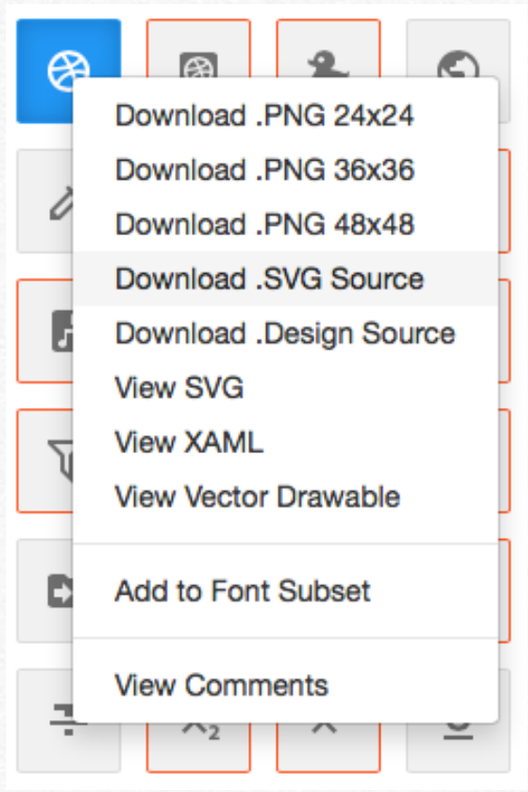
File and Directory Structure

Inside the "js" folder there is a file called app.js and a copy of modernizr.js as well.

Icons

For our page I am using assets from Google's Material Design efforts - specifically the icons. In this case we are using a collection located at http://materialdesignicons.com.

This image displays some of the options available in addition to the font which is downloadable as well.



Building the HTML

1-base.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="initial-scale=1.0">

    <title>My Portfolio</title>
    <link rel="stylesheet" href="css/style.css">
    <script src="js/modernizr.js"></script>
</head>
<body>
    <main class="wrap group">
```

```
        <aside class="sidebar">
           <div class="my-info">

           </div>
           <nav class="menus">

           </nav>
        </aside>
        <div class="content">

        </div>
    </main>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
    <script src="js/app.js"></script>
</body>
```

A simple layout with the main wrapping element containing a sidebar aside and a content div. Inside the sidebar are two sections, my-info and a nav with a class of menus.

2 The Sidebar > my-info

2-sidebar.html - paste into my-info.

```
<img src="images/drunkenPirate_avatar.png" alt="drunken pirate" class="portfolio-image">
<h1>Cap'n Seymour Butts</h1>
<h2>Experienced pirate and leader of the salty seamen operating in the Carribean and South Seas.</h2>
<div class="social group">
    <a href="#" class="dribbble">Dribbble</a>
    <a href="#" class="poopy">Poopy</a>
    <a href="#" class="facebook">Facebook</a>
    <a href="#" class="github">Github</a>
</div>
<!-- </div> -->
```

```
<div class="searchbar">
  <input/>
</div>
```

3 Sidebar Menus

3-sidebarmenus.html - paste into menus

```
<h3 class="work">Work</h3>
<ul>
  <li><a href="#" class="current-menu-item">S.S. Pearl Necklace</a></li>
  <li><a href="#">The Sack of the Innocents</a></li>
  <li><a href="#">Pipe and First Mate</a></li>
</ul>

<h3 class="about">About</h3>
<ul>
  <li><a href="#">Skills</a></li>
  <li><a href="#">Experience</a></li>
  <li><a href="#">Testimonials</a></li>
</ul>

<h3 class="contact">Contact</h3>
<ul>
  <li><a href="#">Address</a></li>
  <li><a href="#">Phone</a></li>
  <li><a href="#">Social Networks</a></li>
</ul>
```

4 Content HTML

4-content.html - inside the .content div

```
<article class="portfolio-item group first">
    <header class="portfolio-info">
```

```
      <div class="title">Call of Booty</div>
      <div class="date">7 Nov 2013</div>
      <div class="description">
        Arrrgh mateys! This be the finest vessel sailing these
seas that yer eyes ever laid sight on. Arrrgh.
      </div>
    </header>
    <figure class="portfolio-image">
      <img src="images/drunkenPirate_thumb3.png" alt="Pirate
Ship">
    </figure>
  </article>

  <article class="portfolio-item group">
    <header class="portfolio-info">
      <div class="title">The Sack of the Innocents</div>
      <div class="date">7 Nov 2013</div>
      <div class="description">
        This be a detail of the sacking of the innocents where
I played an important role as business analyst and in
implementation.
      </div>
    </header>
    <figure class="portfolio-image">
      <img src="images/drunkenPirate_thumb2.png" alt="Sack of
the Innocents">
    </figure>
  </article>

  <article class="portfolio-item group">
    <header class="portfolio-info">
      <div class="title">Pipe and First Mate</div>
      <div class="date">7 Nov 2013</div>
      <div class="description">
        After a hard day of lootin' and shootin' I like to
relax with my best mates and tally up the days booty.
      </div>
    </header>
    <figure class="portfolio-image">
      <img src="images/drunkenPirate_thumb1.png" alt="Pipe and
Mate">
    </figure>
  </article>

  <article class="portfolio-item group loading-wrap"
style="display:none">
    <header class="portfolio-info">
    </header>
    <figure class="portfolio-image">
      <div class="loading">
        <img src="images/loading.png" alt="Loading"
class="rotate"> Loading...
      </div>
    </figure>
  </article>
```

The Terminal

Open the page in a browser using http: $python -m SimpleHTTPServer

Introducing SASS and CSS Processors

Examine minimization of existing css and js files

Go to Sass and Koala page.

Discuss map files.

Try to induce an error.

SASS lets you use features that don't exist in CSS ***yet*** like variables, nesting, mixins, and inheritance.

5-base.scss

Variables

```
$black: #000;
$white: #FFF;
$banana: #fad46b;
$theblackflag: #364347;
$firstblood: #cc0000;
$strawberry: #f84545;
$babyblue: #e1eef3;
$gray: #f7f7f7;
$darkgray: #5e5e5e;
$lightgray: #a1a1a1;
$verylightgray: #b2c6cd;
$landlubber: #28948c;
$overboard: #009dcd;

$padding: 10px;
$margin: 10px;
$main-width: 1000px;

$font-size: 14px;
$font-family: 'Lato', sans-serif;
$line-height: 1.4;

$break-four: 1050px;
$break-three: 760px;
$break-two: 520px;
$break-one: 360px;
```

Custom Font and @import

```
@import url('http://fonts.googleapis.com/css?family=Lato:300,400,700');
@import url('normalize.css');
```

Clearfix

```
//clearfix - see http://nicolasgallagher.com/micro-clearfix-hack/
.group:after {
  content: "";
  display: table;
  clear: both;
}
```

6 Main (default) CSS

6-main.scss - relative positioning, box sizing, vendor prefixes

```
//Main styles
* {
    position: relative;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    -ms-box-sizing: border-box;
    box-sizing: border-box;
}
html, body, .wrap {
    min-height: 100%;
}
```

Variable use

```
body {
    color: $theblackflag;
    font-size: $font-size;
    font-family: $font-family;
    line-height: $line-height;
    background: $banana;
}
```

7 Sidebar CSS

7-sidebar.scss - sass nesting, sass * operator. Responsive design.

```
.sidebar {
```

```
    width: 100%;
    height: 100%;

    @media screen and (min-width: $break-three) {
        float: left;
        width: 20%;
        background: url(../images/hero.png) bottom center no-
repeat;
        background-size: 80%;
        padding-bottom: 14%;
    }
    .my-info {
        text-align: center;
        padding: $padding*3 0;
        .portfolio-image {
            border-radius: 50%;
        }
        h1, h2 {
            font-weight: normal;
        }
        h1 {
            font-size: 120%;
        }
        h2 {
            font-size: 100%;
        }
    }
}
.searchbar {
    text-align: center;
}
```

8 Social CSS

8-social.scss - nesting, responsive design and images, referencing parent selectors

```
.social {
    width: 120px;
    margin: 0 auto;

    a {
        float: left;
        width: 25px;
        height: 25px;
        margin: 0 $margin/5;
        background: url(images/social-media.png) no-repeat;
        text-indent: -9999px;

        @media screen and (-webkit-min-device-pixel-ratio: 2),
screen and (min-device-pixel-ratio: 2) {

            background: url(images/social-media@2x.png)
             no-repeat;
            background-size: 101px 51px;
        }

        &.dribbble {
            background-position: 0px 0px;

            &:hover {
                background-position: 0px -25px;
            }
        }
        &.twitter {
            background-position: -25px 0px;

            &:hover {
                background-position: -25px -25px;
            }
        }
        &.facebook {
            background-position: -50px 0px;
```

```
            &:hover {
                background-position: -50px -25px;
            }
        }
        &.googleplus {
            background-position: -75px 0px;

            &:hover {
                background-position: -75px -25px;
            }
        }
    }
}
```

9 Menu lists

9-sidebarlists.scss - transitions, operators, .open class is for later

```
ul {
    list-style: none;
    padding: 0;
    display: none;
    margin: 0 $margin*13.5 $margin;
    text-align: left;

    @media screen and (min-width: $break-three) {
        margin: 0 0 0 $margin*2.5;
        display: block;
        width: auto;
    }
    &.open {
        display: inline-block;
        margin: 0 auto $margin $margin*6;

        @media screen and (min-width: $break-three) {
            margin: 0 0 0 $margin*2.5;
            display: block;
            width: auto;
        }
    }
    li {
        a {
            color: $theblackflag;
            text-decoration: none;
            -webkit-transition: color 0.4s ease;
            -moz-transition: color 0.4s ease;
            -o-transition: color 0.4s ease;
            -ms-transition: color 0.4s ease;
            transition: color 0.4s ease;

            &:hover, &.current-menu-item {
                color: $strawberry;
                -webkit-transition: color 0.4s ease;
                -moz-transition: color 0.4s ease;
                -o-transition: color 0.4s ease;
                -ms-transition: color 0.4s ease;
                transition: color 0.4s ease;
            }
        }
    }
}
```

10 Left side menus and Icon Sprites

10-sidebarmenus.scss

Pointer on the H3 is for responsive part. :before pseudoclass for icons.

```
.menus {
    text-align: center;

    @media screen and (min-width: $break-three) {
        padding: $padding*2 $padding*3;
    }
```

```
h3 {
    text-transform: uppercase;
    font-size: 120%;
    font-weight: normal;
    padding-left: $padding*2.5;
    cursor: pointer;
    width: 20%;
    margin: $margin*2 $margin*11 $margin;

    @media screen and (min-width: $break-one) {
        margin: $margin*2 auto $margin*2;
    }

    @media screen and (min-width: $break-three) {
        margin: $margin*2 0 $margin 0;
    }

        &:before {
            content: "";
            position: absolute;
            top: 0px;
            left: 0px;
            height: 18px;
            display: block;
            background: url(images/nav-sprite.png) no-repeat;

            @media screen and (-webkit-min-device-pixel-ratio:
2), screen and (min-device-pixel-ratio: 2) {
                background: url(images/nav-sprite@2x.png) no-
repeat;
                background-size: 49px 18px;
            }
        }

        &.work {
            color: $strawberry;

            &:before {
                width: 15px;
                background-position: 0px 0px;
            }
        }

        &.about {
            color: $landlubber;

            &:before {
                width: 17px;
                background-position: -15px 0px;
            }
        }

        &.contact {
            color: $overboard;

            &:before {
                width: 17px;
                background-position: -32px 0px;
            }
        }
    }
}
```

11 Main content

11-content.scss - will cause an error. Note the use of :before to add dot

```
.content {
    width: 100%;
    min-height: 100%;
    background: $white;
```

```
@media screen and (min-width: $break-three) {
    float: left;
    width: 80%;
}

.portfolio-item {
    background: $babyblue;

    &:before {
        content: "";
        position: absolute;
        width: 3px;
        background: darken($verylightgray, 5%);
        top: 0px;
        left: 17px;
        bottom: 0px;
    }

    &.first {
        &:before {
            top: 30px;
        }
    }

    .portfolio-info {
        min-height: 100%;
        color: $darkgray;
        padding: $padding*2 $padding*2 $padding*2
$padding*4;
        -webkit-box-sizing: border-box;
        -moz-box-sizing: border-box;
        -ms-box-sizing: border-box;
        box-sizing: border-box;

        @media screen and (min-width: $break-three) {
            float: left;
            width: 80%;
        }

        .date {
            font-size: 100%;
            color: $theblackflag;
        }

        .title {
            font-size: 110%;
            color: $theblackflag;
            margin-bottom: $margin;

            &:before {
                content: "";
                position: absolute;
                width: 11px;
                height: 11px;
                border-radius: 50%;
                border: 2px solid $gray;
                background: $strawberry;
                left: -29px;
                top: 3px;
            }
        }

    }

}
```

12 Inside content styles at end

12-portfolioimage.scss - may cause error.

```
.portfolio-image {
    padding: $padding*2;
```

```
background: $white;
border-left: 1px solid darken($gray, 10%);
text-align: center;

@media screen and (min-width: $break-three) {
    float: left;
    width: 70%;
}

img {
    width: 100%;
    max-width: 610px;
    height: auto;
}

div.loading {
    img {
        width: auto;
        height: auto;
        }
    }
}
```

13 Spinning Load Icon

13-loadingicon.scss

CSS animation - absolutely positioning the loading div and making sure it is centered by using a margin-left of -50px, which is half of the element's width. This is to make up for the fact CSS positions elements from the top left corner.

In the HTML we assigned a class of rotate to the loading <img> tag which we will use as the hook to perform a CSS animation. We want a 360 degree rotation that never stops.

```
div.loading {
    color: darken($gray, 20%);
    position: absolute;
    width: 100px;
    bottom: 15px;
    left: 50%;
    margin-left: -50px;
// can use calc here margin-left: calc(100px/2)
// often used with percentages width: calc(100% - 80px);

    img {
        vertical-align: middle;

        &.rotate {
// -webkit required on some
            animation-name: rotate;
            animation-duration: 1s;
            animation-iteration-count: infinite;
            animation-timing-function: linear;
        }
    }

  // next partial here
}
```

14 Keyframes

14-loadingicon-spin.scss

Keyframes - setting key points in the animation to change the element. transform-origin property tells CSS where the centre point for the animation is.

```
@-webkit-keyframes rotate {
    100% {
        transform: rotate(360deg);
        transform-origin: center center;
```

```
    }
}

@keyframes rotate {
    100% {
        transform: rotate(360deg);
        transform-origin: center center;
    }
}
```

15 jQuery

15-app.js

✦ jQuery ready event http://api.jquery.com/ready/

✦ function(e) http://stackoverflow.com/questions/3535996/jquery-javascript-functione-what-is-e-why-is-it-needed-what-does-it-ac

```
$(function() {
    $('.menus h3').on('click', function(e) {
     console.log('made it');
        $(this).next('ul').toggleClass('open');
        return false;
    });
});
```

Control Bar (Quark)

```
$(document).ready(function(){
    // console.log('test');
    $(".toggleImage").on("click", function(){
        $("figure.media-primary--showcase").toggleClass("hide");
        // console.log('test');
    });

    $(".toggleBullets").on("click", function(){
        $(".bulletted").toggleClass("hide");
        $(".notbulletted").toggleClass("hide");
        // console.log('test');
    });

    $(".addBullets").on("click", function(){
        $(".bulletted").addClass("show");
        $(".notbulletted").toggleClass("show");
        // console.log('test');
    });

    $(".toggleImageSize").on("click", function(){
        // console.log('test');
        $("figure").toggleClass("media-primary--showcase");
        $(".content__meta-container").toggleClass("content__meta-container--showcase");
    });
```

```
// breakpoint calc
//min-width
var breakpoints = [
{breakfive: "1300px" },
{breakfour: "1140px" },
{breakthree: "980px" },
{breaktwo: "740px" },
{breakone: "360px" }
];


var $window = $(window);
var width = $window.width();
checkWidth();

$(window).resize (function(){
    width = $window.width();
    checkWidth();
});
function checkWidth(){
    if (width >= 1300){
        $(".showBreakpoint").val("$break-five: 1300px or higher. Current: " + $window.width());
    } else if (width >= 1140) {
        $(".showBreakpoint").val("$break-four: 1140px or higher. Current: " + $window.width());
    } else if (width >= 980) {
        $(".showBreakpoint").val("$break-three: 980px or higher. Current: " + $window.width());
    } else if (width >= 740) {
        $(".showBreakpoint").val("$break-two: 740px or higher. Current: " + $window.width());
    } else if (width >= 360) {
        $(".showBreakpoint").val("$break-one: 360px or higher");
    };
};
```

16 Infinite Scroll

16-infinite.js - in app.js.

✦ Demonstration of document height

```
$(window).on('resize', function(e) {
   console.log('this is ' + $(document).height() + ' high.');
});
```

✦ Demonstration of window height

```
$(window).on('resize', function(e) {
   console.log('the window is ' + $(window).height() + '
high.');
});
```

✦ main code

```
var visibleHeight = $(document).height() - $(window).height();
var items;
```

visibleHeight grabs the window height and subtracts that from the overall document height leaving us with *the height of the area currently visible in the users' browser.*

```
storeElements();

$(window).on('resize', function(e) {
    updateHeight();
});

$(window).on('scroll', function(e) {
    loadContent();
});
```

Above are a few function calls to functions that don't exist yet, we create them below.

```
function storeElements() {
    items = $('.portfolio-item:lt(3)').clone();
    //Strip the .first class from selection
    items.removeClass('first');
}
```

The **storeElements** function serves as a way to populate our items variable with some DOM elements. In our case we want to grab the first three portfolio-items. The use of jQuery's lt(x) allows us to selectively pick the elements we want.

Once we have the selection we clone( ) them, so we aren't using the actual elements but a copy instead. The last step is to remove the 'first' class (if it exists), as none of our new elements are the first in the list.

Place the following function above *storeElements*. This is probably the simplest function as it is only doing what we did on *document.ready*. The reason I have used a function to do this is to keep it reusable.

```
function updateHeight() {
   // fires when the new content is appended
  visibleHeight = $(document).height() - $(window).height();
}
```

The function that is doing the real work:

```
function loadContent() {
    if($(window).scrollTop() >= visibleHeight) {
        $(window).unbind('scroll');
        var loadingWrap = $('.loading-wrap');
        loadingWrap.fadeIn(function() {
            setTimeout(function() {
                loadingWrap.before(items);
                loadingWrap.hide(function() {
                    updateHeight();
                    storeElements();
                    $(window).on('scroll', function()
{ loadContent(); });
                });
```

```
            }, 500);
        });

    }
}
```

- ✦ Check if the scroll position is more than (scrolled past) or equal to (currently at) visibleHeight.
- ✦ If it is, remove the scroll event handler from the window so we can process the content.
- ✦ Cache the loading-wrap for use later.
- ✦ Fade loading-wrap in and once the fade completes...
- ✦ ...set a small Timeout to simulate "loading" the content.
- ✦ Attach our cloned items before the loading-wrap. This will slot in nicely between the loading icon and the current portfolio-items.
- ✦ Hide loading-wrap and, once hidden, updateHeight, storeElements and re-attach the scroll event to the window with instructions to run this function again.

We are fading in the loading-wrap which is actually already visible in our HTML. Fix that by adding an inline style to that element.

17-infinite.html

```
<div class="portfolio-item group loading-wrap"
style="display:none;">
```

and test.

Inside the left menu h3 click handler we created right at the start we must add in a call to updateHeight(), this way when we click to open a menu the variables are updated to reflect the change in height of the document. Without this, if you open a menu on mobile the "loading" functionality would be broken as the visibleHeight variable would be incorrect.

18-infinite-sidemenufix.js

```
$('.menus h3').on('click', function(e) {
    $(this).next('ul').toggleClass('open');
    updateHeight();
    e.preventDefault(); return false;
});
```

jQuery

Lorem ipsum dolor rutur amet. Integer id dui sed odio imperd feugiat et nec ipsum. Ut rutrum massa non ligula facilisis in ullamcorper purus dapibus.

jQuery

Intro

1. Setting up our first jQuery page

1 First Pass

Include a link to the jQuery library and set up an event listener when the DOM of the page is loaded, and inside the event handler we'll use jQuery to insert some content into the webpage, indicating that jQuery has properly loaded, and is working correctly.

Open 1-start.html and ExampleSnippets.txt.

2 Selectors and Filters

Filters are used to refine the results that come back from selectors.

2-SelectandFilter.html - un-comment line 1-6

Selectors work using a CSS-like syntax. $(“p”) will select all of the paragraph tags and will return them as a list that you can then further operate on using a variety of operations.

3 Create and Change

3-createandChange.html

4 Events

4-events.html

Accordion

✦ Add CSS to the provided HTML

```
* {
   margin:0;
   padding:0;
}
body { font: 12px/1.5 Arial, Helvetica, sans-serif }
.drawers {
   list-style:none;
```

```
}
.drawer-handle {
   cursor: pointer;
   color: #fff;
   background: url('images/accordion_toggle.jpg') top left;
   border-top: 2px solid #fff;
}
.drawer-handle + ul {
   display: none;
}
.selected {
   background-image: url('images/accordion_toggle_active.jpg');
}
```

The JavaScript

```
$(document).ready(function(){
   $('li.drawer ul').hide();
});
```

~~~~~~~~~~~~~~~~

```
$(document).ready(function(){
   $('li.drawer ul').hide();
   $('li.drawer ul:first').show();
});
```

~~~~~~~~~~~~~~~~

```
$(document).ready(function(){
   $('li.drawer ul').hide();
   $('li.drawer ul:first').show();
   $('h2.drawer-handle').click(function(){
   });
});
```

~~~~~~~~~~~~~~~~

```
$(document).ready(function(){
   $('li.drawer ul').hide();
   $('li.drawer ul:first').show();
   $('h2.drawer-handle').click(function(){
      $('li.drawer ul').slideUp('slow');
      $(this).next().slideDown("slow");
   });
});
```

✦ Add background graphic changes

```
$(document).ready(function(){
   $('li.drawer ul:first').show();
   $('li.drawer h2:first').addClass('selected');
   $('h2.drawer-handle').click(function(){
      $('li.drawer h2').removeClass('selected');
      $('li.drawer ul').slideUp('slow');
      $(this).next().slideDown("slow");
      $(this).addClass('selected');
   });
});
```

✦ Tweak the CSS

```
.drawers {
   list-style:none;
   width: 100%;
}
.drawer-handle {
   cursor: pointer;
   color: #fff;
   background: #AAD06B url('images/accordion_toggle.jpg') top
right no-repeat;
   border-top: 2px solid #fff;
}
...
.selected {
   background-image: url('images/accordion_toggle_active.jpg');
   background-color: #E05530;
}
```

Responsive Design
~~~~~~~~~~~~~~~~

```
.drawers {
   list-style:none;
   float: left;
   width: 50%;
}
.boxee {
   width: 50%;
   height:20em;
   float:right;
   background-color: #E05530;
}
   <div class="boxee">

   </div>
```

Add responsiveness

```
.drawers {
   list-style:none;
   width: 100%;
}
...
.boxee {
   width: 100%;
   height:20em;
   background-color: #E05530;
}
@media screen and (min-width: 760px) {
   .boxee {
      float: left;
      width: 50%;
   }
   .drawers {
      float: left;
      width: 50%;
   }
}
```

Box model

```
.boxee {
   border: 3px solid #fff;
   color: #fff;
   width: 100%;
   min-height:10em;
   background-color: #E05530;
}
   * {
      margin:0;
      padding:0;
      box-sizing: border-box;
   }
```

Convert to SASS

✦ Note the use of nested media queries and compare the syntax of CSS and compiled CSS

```
.drawers {
   list-style:none;
   width: 100%;

   @media screen and (min-width: 760px) {
      float: left;
      width: 50%;
   }
}

.boxee {
   border: 3px solid #fff;
   color: #fff;
   width: 100%;
   min-height:10em;
   background-color: #E05530;

   @media screen and (min-width: 760px) {
```

```
      float: left;
      width: 50%;
    }
  }
```

The problem with jQuery

```
$(document).ready(function(){

  var currentStep = 0;
  $('#step1').hide();
  $('#step2').hide();

  $('#btnStep1').click(function(){
    $('#step1').show();
    $('#step2').hide();
    currentStep = 1;
  });

    $('#btnStep2').click(function(){
    $('#step1').hide();
    $('#step2').show();
    currentStep = 2;
  });

});
```

GIT

3

Git

As a precursor run through the steps on the github site tutorial http://try.github.com/.

Git Config

```
git config
git config --global user.name "DannyBoyNYC"
git config --global user.email "daniel.deverell@gmail.com"

git config --list
```

mkdir

```
mkdir repos
mkdir repos/git-basics
cd repos/git-basics
git init
Initialized empty Git repository in /Users/Home/Desktop/repos/
git-basics/.git/
```

Examine the .Git Directory

```
ls -al
cd .git
ls
HEAD    config    hooks     objects
branches  description  info    refs
```

Git Status

✦ NB - navigate into the repo/repo-name before running status.

```
git status
On branch master
Initial commit
nothing to commit (create/copy files and use "git add" to
track)
```

✦ Git doesn't auto track files - only those you tell it to. Adding files creates untracked files. Create and add note.txt to git-basics and run status.

```
git add *.txt
git status
git commit
```

✦ By default this goes into VIM. Type in “Initial Commit” at the top. Hit ESC and type “:wq” to write the file and exit.

```
[master (root-commit) cb26b70] Initial Commit class two
 44 files changed, 13746 insertions(+)
 create mode 100755 jQuery/1-end.html
 ...
 create mode 100644 site/js/modernizr.js

git status
On branch master
nothing to commit, working directory clean
```

✦ Add note2.txt text file and run status

```
git status
On branch master
...

Daniel$ git add note2.txt
git status
```

```
git add note2.txt
```

✦ Commit without going into VIM and add a quick message

```
git commit -m "Added readme file"
[master 05c1395] Added readme file
 1 file changed, 2 insertions(+)
 create mode 100644 readme.txt
```

Modified Files

✦ Make a change to index.html and run status

✦ See what's changed in the file run

```
git diff
```

✦ Its un-staged Run git add and then git status. Notes the results. Run git diff again.

```
git add .
git status
git diff --staged
git commit -m "message"
```

✦ N.B. You can set up a third party app to run diff

Pirate Fun

Mobile first strategy in the SCSS.

```
.sidebar {
  width: 100%;
  height: 100%;
  padding-bottom:0;
  background-image: none;

  @media screen and (min-width: $break-three) {
    float: left;
    width: 20%;
    padding-bottom:14%;
    background: url(../images/hero.png) bottom center no-repeat;
    background-size: 80%;
  }
```

Branching

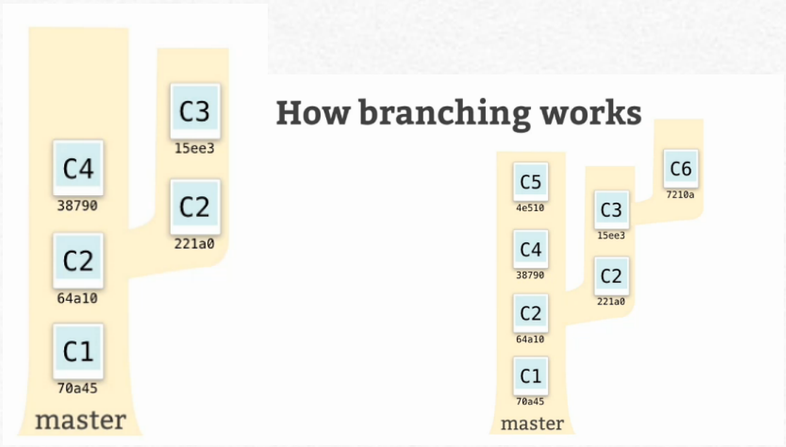
✦ Allows you to experiment without changing the main project.

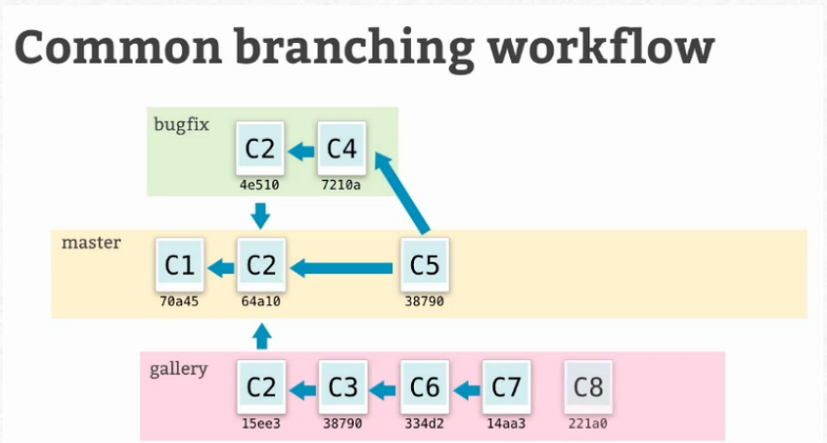
✦ Create a branch

```
git branch addfonts
```

✦ We need to be on the branch. Do not assume you are on the new branch. Creates branch based on current location.

```
git status
On branch master
nothing to commit, working directory clean
```





✦ Branch Switching

```
git checkout addfonts
```

```
git status
On branch addfonts
nothing to commit, working directory clean
```

```
git checkout master
Switched to branch 'master'
```

✦ Create, checkout and switch to the branch immediately. Careful not to branch from addfonts. Note - for the purposes of this tutorial bugFix must be created at the same time as addfonts.

```
git checkout -b bugFix
Switched to a new branch 'bugFix'
```

✦ Check which branch you are on:

```
git branch
  addfonts
* bugFix
  master
```

Pushing Files to Remote Repos - Github

Create a remote repo on githhub (name doesn't have to match), add description, gitignore (try Wordpress to start) and license.

Modify Readme file adds description to repo. Use text editor in Github. Stores in md format. See github help for md and github flavored md.

Copy URL from github.

```
git remote add origin https://github.com/.../...
git push -u origin master
```

-u means upstream, links the 2 branches

Enter username and password for Github.

If problems arise:

```
git push -f origin master
git status
```

Refresh remote repo. Try editing the files in github editor (Zen mode).

Github

Create github basics remote repo and add description, gitignore, license. Clone the repo locally by clicking on Clone button to open github client.

```
git pull [url]
```

git log

git pull origin master

git diff HEAD

========

Updating Branches with Fast Forward Commits

✦ On the addfonts branch

```
git checkout addfonts
```

✦ Add fonts folder and add changes to CSS. Update index as follows.

```
<h1>Cap'n Seymour Butts</h1>
to
<h1><span aria-hidden="true" data-icon="&#xE087;"
class="github-icon">Cap'n Seymour Butts</span></h1>
```

✦ Copy and paste from fonts.txt at the top. Add the _fonts folder.

✦ Remember to save and close files when switching branches.

```
git add .
git status
git commit -m "added fonts and info to h1"
git checkout master
```

✦ Watch the finder to see the switch. We have changes to both the files and the file structure.

Merging 1 - fast forward

✦ Powerful and complicated esp. with many branches and changes. Directories must be clean with no pending changes or untracked files. Check both master and addfonts using status.

✦ On addfonts

```
git branch master
git status
git branch master
git status
```

✦ NB Must be on the branch you want to merge to

```
git merge addfonts
git status
```

Git looks at the most common ancestor to merge. If there are no changes to master branch then we get a fast forward merge.

After this merge the master and addfonts are exactly the same.

Complex or Recursive Merge

✦ bugFix was created before the merge so is missing additions. What happens when we try incorporating changes made to bugFix into the master branch?

```
git checkout bugFix
```

✦ Note the finder window.

✦ Make a change to the CSS (nav headers in small screen mode).

```
git status
```

✦ -a all changes, -m message

```
git commit -a -m 'fixed a bug'
git status

git checkout master
git status
git merge bugFix
```

This is not a fast forward merge as it brings in just the changes needed. You are making a new commit. In VIM add a comment at the top and type ESC :wq

Recursive merges pull in the latest changes to both branches. It leaves the changes from addfonts alone.

Resolving Conflicts

When you edit a file in both branches you want to merge which edit does Git keep? Esp. if its in the same line.

```
clear
git branch
git status
git checkout bugFix
```

Make a change in index.html (to the same h1 that was edited in addfonts branch). Add a few other edits for good measure.

This will be an issue because the current h1 is from addfonts

```
git commit -a -m 'changed html and h1'

git checkout master
git status
git merge bugFix
```

✦ This should produce a merge conflict which you need to fix manually. Open index in editor and delete the unwanted code carefully. eg:

```
<<<<<HEAD
==========
>>>>>> bugFix

git status
git commit -a -m 'resolving merge conflicts'
git status
```

Managing Branches

```
git branch
```

Track merge activity after a long time

```
git branch --merged
```

Shows all branches already merged at some point

```
git branch --no-merge
```

Shows unmerged branches

```
git branch -d addfonts
```

Message possible - you haven't merged it yet, can't delete

```
git branch -D bugFix
```

Angular

4

On Github

On Anjular JS .org

Angular

Understand what you need to know before beginning 1.6

✦ JavaScript debug console (advanced in Safari)

✦ JavaScript types using typeof in the console

```
typeof 1234
typeof 1234.56  // number
typeof  10/3  // NaN
typeof true  // boolean
typeof false  // boolean
typeof “cat”
typeof ("cat " + "meow")  // string
typeof { user: "Daniel" }   // object
typeof function(){ return 5; }  // function
typeof [ 1, 2, 3, 4 ]  // arrays are objects
Array.isArray([ 1, 2, 3 ])  // true
Array.isArray({ user: "Daniel" })  // false
var x = function() { return 5; }
x()
```

Angular

✦ Bundles many JS utilities into a single entity

✦ Declarative / Descriptive vs Awkward / Invasive.

✦ Like using HTMLX e.g. 6.0

Write a hello world angular page in a new folder within the existing repo.

```
<html ng-app>
<head>
  <script src="https://code.angularjs.org/1.3.0-rc.1/
angular.min.js"></script>
</head>
<body>
  Everybody wanna shout "{{ "Hello World" }}"
</body>
</html>
```

✦ ng-app is a directive (turns Angular on - i.e. angular looks for directives to know what to do). See also data-ng-app and validation.

✦ *Mustaches* evaluate expressions

Download Bootstrap (for SASS)

Using bootstrap and SASS from http://getbootstrap.com/getting-started/

✦ See also components and JavaScript reference. Need JS, CSS and Fonts only for now. Work with Alerts - blue one.

✦ Move files and edit the html to include bootstrap functionality

```
<html ng-app>
<head>

<link rel="stylesheet" type="text/css" href="https://
netdna.bootstrapcdn.com/bootstrap/3.2.0/css/
bootstrap.min.css">

<script src="https://code.angularjs.org/1.3.0-rc.1/
angular.min.js"></script>
</head>
<body style="padding: 50px">
  <div class="alert alert-success">
```

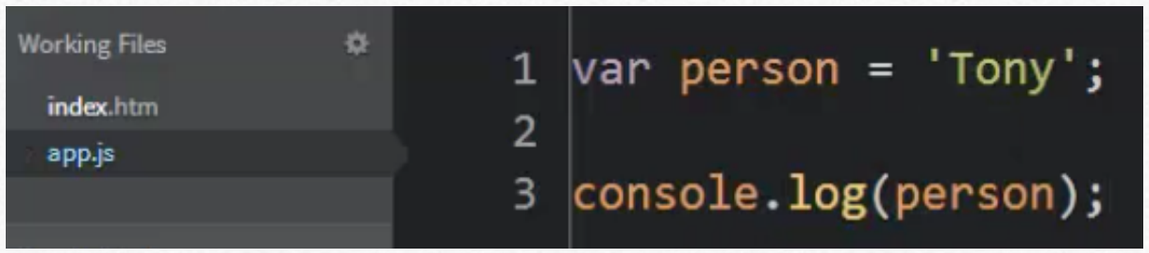
```
    Everybody wanna shout "{{ "Hello World" }}"
  </div>
</body>
</html>
```

ng-model and ng-init Directives 1.5?

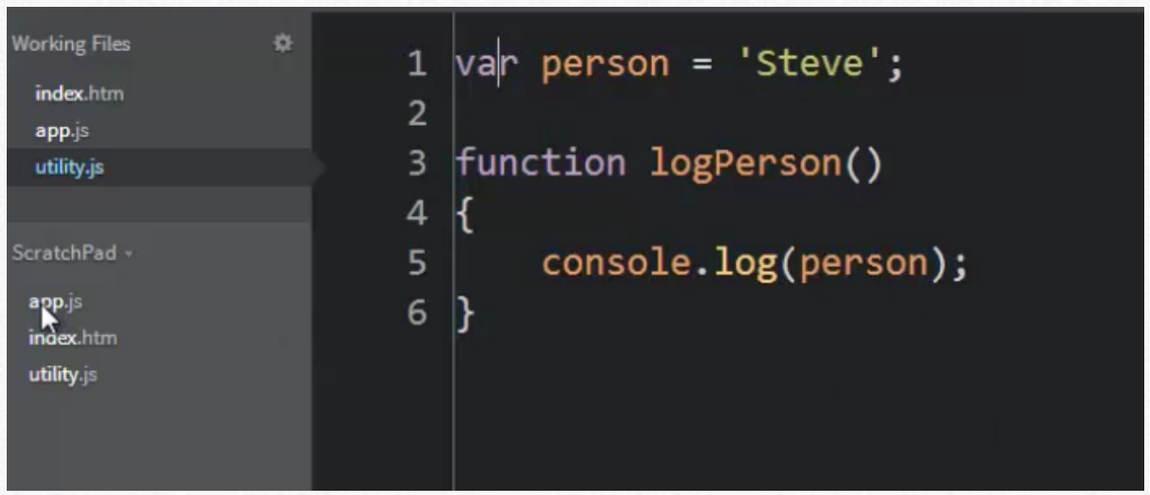
Binds the value of an element to a variable. Uppercase is a filter.

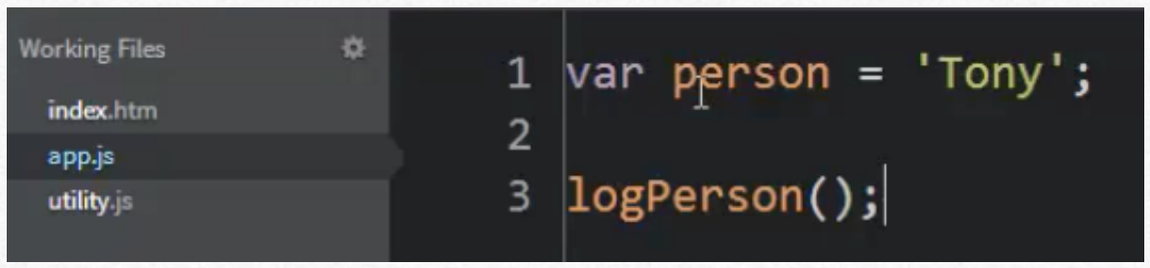
```
<html ng-app>
...
</head>
<body style="padding: 50px" ng-init="messageText = 'Hello
World!'">
  <div class="alert alert-success">
    <input ng-model="messageText" size="30"/><br/>
    Everybody wanna shout "{{ messageText | uppercase }}"
  </div>
</body>
</html>
```

- ✦ Use an ***object*** for the ng-init directive (see: http://www.w3schools.com/js/js_objects.asp)
- ✦ Encapsulation in order not to pollute the global namespace. Variables and functions do not over write variables and functions defined elsewhere. (See global_namespace directory)



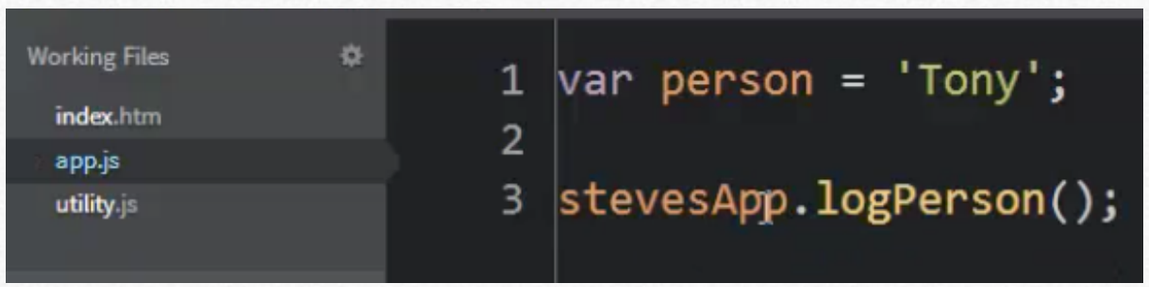
person is in the global namespace





Using Steve's code returns Tony. Variable has been overwritten. Bad for reusable code.

Create an empty object and then encapsulate / add namespace.




```
<html ng-app>
...
</head>
<body style="padding: 50px" ng-init="greeting = { greeter:
'Daniel' , message: 'Hello World' }">
  <div class="alert alert-success">
    <input type="text" ng-model="greeting.greeter" size="30"/>
    <input type="text" ng-model="greeting.message" size="30"/>
    {{greeting.greeter }} says "{{ greeting.message }}"
  </div>
</body>
</html>
```

The ng-repeat Directive

✦ Use an array for the ng-init directive

✦ Use ng-repeat to display results

```
<html ng-app>
...
</head>
<body style="padding: 50px">

<div ng-init="portfolios = ['Call of Booty', 'The Sack of the
Innocents', 'Pipe and First Mate']" >
<ul>
  <li ng-repeat="portfolio in portfolios">
    {{ portfolio }}
  </li>
</ul>
</div>
</body>
</html>
```

Angular API Reference and Filters

Examine the directive topic. ngApp vs ng-app - feels natural in the appropriate context.

✦ expand on the previous html to use ***objects in an array***

```
<html ng-app>
...
</head>

<body style="padding: 50px">
```

```
<div ng-init="portfolios = [
{ name: 'Call of Booty', date: '2013-09-01' },
{ name: 'The Sack of the Innocents', date: '2014-04-15' },
{ name: 'Pipe and First Mate', date: '2012-10-01' } ]"
>

<p>Filter list: <input ng-model="searchFor" size="30"/></p>

<ul>
  <li ng-repeat="portfolio in portfolios |
    filter:searchFor |
    orderBy:'date' "
  >
      {{ portfolio.name  }}

  </li>
</ul>

 <p> There are {{ portfolios.length }} portfolios available to
view.</p>
</div>
</body>
</html>
```

ngClass the index (NEEDS UPDATING see 2.end)

✦ add the appropriate CSS

```
<style>
  .even { color: red; }
  .odd { color: blue; }
</style>
```

✦ update the array of objects

```
<div ng-init="portfolios = [
    {name: 'vessel1309', title: 'Call of Booty', date:
'2013-09-01', description: 'Arrrgh mateys! This be the finest
vessel sailing these seas that yer eyes ever laid sight on.
Arrrgh.', imageurl: 'images/drunkenPirate_thumb3.png' },
    {name: 'innocents1404', title: 'The Sack of the
Innocents', date: '2014-04-15', description: 'This be a detail
of the sacking of the innocents where I played an important
role as business analyst and in implementation.', imageurl:
'images/drunkenPirate_thumb2.png' },
    {name: 'firstmate1210', title: 'Pipe and First Mate',
date: '2012-10-01', description: 'After a hard day of lootin\'
and shootin\' I like to relax with my best mates and tally up
the days booty.', imageurl: 'images/
drunkenPirate_thumb1.png' }
    ]" >

<p> There are {{ portfolios.length }} portfolios available to
view.</p>
<ul>
  <li ng-repeat="portfolio in portfolios | filter:searchFor |
    orderBy:'date' "
    ng-class="{ even: $even, odd: $odd }" >
{{ $index + 1 }} = {{ portfolio.title }}
( {{ portfolio.date }} )
  </li>
</ul>
```

✦ Also - can use a repeat in repeat to expose the key value pairs

```
<ul>
<li ng-repeat="portfolio in portfolios">
  <ul>
    <li ng-repeat="(key, value) in portfolio">
     <strong>{{key}}</strong> - {{value}}
    </li>
  </ul>
</li>
</ul>
```

Using Bootstrap to Create Panels

```
<body style="padding: 50px">

  <div class="container-fluid" ng-init="portfolios = [...]">
  <p>Filter list: <input ng-model="searchFor" size="30"/></p>

  <div class="col-xs-4" ng-repeat="portfolio in portfolios |
filter:searchFor | orderBy:'date' " >
    <div class="panel panel-default">
      <div class="panel-heading">
        {{ portfolio.name }} ({{ portfolio.date }})</div>
        <div class="panel-body">
          <p>{{ portfolio.description }}</p>
          <img class="img-responsive" ng-src="images/
{{ portfolio.imageurl }}" />
        </div>
      </div>
    </div>
  </div>

  <div style="clear: left"></div>
  <p> There are {{ portfolios.length }} portfolios so far!</p>
</body>
```

✦ Add Images

```
<div class="panel-heading">
      {{ portfolio.name }} ({{ portfolio.date }})</div>
    <div class="panel-body">
      <p>{{ portfolio.description }}</p>
      <img class="img-responsive"
src="{{ portfolio.imageurl }}" />
    </div>
  </div>
```

✦ Add filters

```
<p>Filter list: <input ng-model="searchFor" size="30"/></p>

<div style="width: 300px; float: left; margin-right: 20px"
  ng-repeat="portfolio in portfolios | filter:searchFor "
  class="panel panel-default">
```

✦ add orderBy date

```
<div style="width: 300px; float: left; margin-right: 20px"
  ng-repeat="portfolio in portfolios | filter:searchFor |
orderBy:'date' "
  class="panel panel-default">
```

ngSrc - Sample Directive

https://docs.angularjs.org/api/ng/directive/ngSrc

✦ Note the error in the console if used without ng-src

```
<img class="img-responsive" ng-
src="{{ portfolio.imageurl }}" />

ng-src="images/{{ portfolio.imageurl }}"
```

A Proper Angular Template

✦ see end files

JavaScript Objects

Object = name / value pair

An object is a name value pair and can contain other objects.

For example:

```
Address:
    {
        Street: ‘Main’,
        Number: 100
        Apartment:
        {
            Floor: 3,
            Number: 301
        }
    }
```

✦ or

```
   var data = {
        "portfolios": [{
            name: 'vessel1309',
            title: 'Call of Booty',
            date: '2013-09-01',
            description: 'Arrrgh mateys! This be the finest
vessel sailing these seas that yer eyes ever laid sight on.
Arrrgh.',
            imageurl: {
                img1: 'images/drunkenPirate_thumb1.png',
                img2: 'images/drunkenPirate_thumb2.png',
                img3: 'images/drunkenPirate_thumb3.png',
            }
        }, {
            name: 'innocents1404',
            title: 'The Sack of the Innocents',
            date: '2014-04-15',
            description: 'This be a detail of the sacking of
the innocents where I played an important role as business
analyst and in implementation.',
```

```
            imageurl: {
                img1: 'images/drunkenPirate_thumb2.png',
                img2: 'images/drunkenPirate_thumb1.png',
                img3: 'images/drunkenPirate_thumb3.png',
            }
        }, {
            name: 'firstmate1210',
            title: 'Pipe and First Mate',
            date: '2012-10-01',
            description: 'After a hard day of lootin\' and
shootin\' I like to relax with my best mates and tally up the
days booty.',
            imageurl: {
                img1: 'images/drunkenPirate_thumb3.png',
                img2: 'images/drunkenPirate_thumb1.png',
                img3: 'images/drunkenPirate_thumb2.png',
            }
        }]
    }

document.getElementById("placeholder").innerHTML =
data.portfolios[0].title + " // " +
data.portfolios[0].description + " // " +
data.portfolios[0].name;
```

✦ variant 1

```
    var output = "<ul>";
    for (var i in data.portfolios) {
        output +=
        "<li>" +

        "<img src=" +
        data.portfolios[i].imageurl.img1 + "/>" +

        "<img src=" +
        data.portfolios[i].imageurl.img2 + "/>" +

        "<img src=" +
        data.portfolios[i].imageurl.img3 + "/>" +

        data.portfolios[i].title + " -- " +
        data.portfolios[i].date + " -- " +
        data.portfolios[i].description  +
        "</li>";
    }
    output += "</ul>";
    document.getElementById("placeholder").innerHTML = output;
```

✦ variant 2

```
 <script src="http://code.jquery.com/jquery-1.7.1.min.js"></
script>

<script>
$.getJSON('portfolios.json', function(data) {
    var output = "<ul>";
    for (var i in data.portfolios) {
        output +=
        "<li>" +

        "<img src=" +
        data.portfolios[i].imageurl.img1 + "/>" +

        "<img src=" +
        data.portfolios[i].imageurl.img2 + "/>" +

        "<img src=" +
        data.portfolios[i].imageurl.img3 + "/>" +

        data.portfolios[i].title + " -- " +
        data.portfolios[i].date + " -- " +
        data.portfolios[i].description  +
```

```
        "</li>";
    }
    output += "</ul>";
    document.getElementById("placeholder").innerHTML = output;
});
</script>
```

MVC

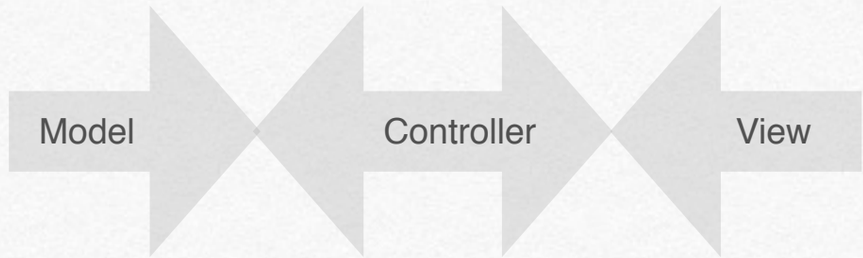
Goals

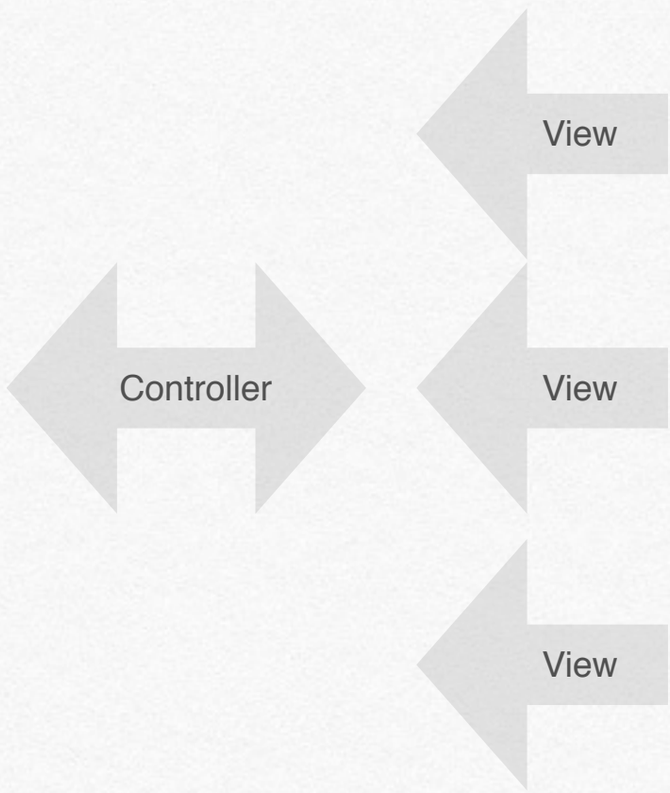
1. **Understand how Angular organizes MVC**
2. **Separate our presentation layer from the code behind it**
3. **Where our data comes from and where to store it**

BUG ALERT - use the 1.2.x version

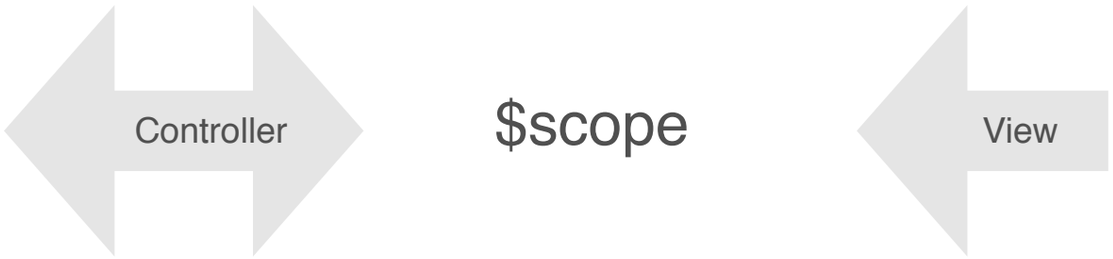
In this section...

- ✦ interface > View
- ✦ backend (organizes data) > Model
- ✦ middleware communicating between the two > Controller (can have multiple views per)



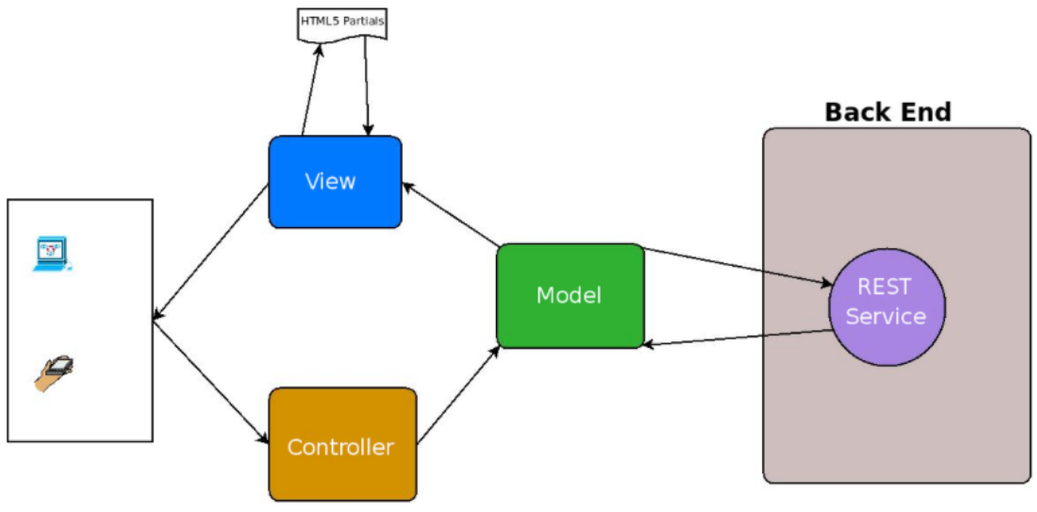


What Connects Views and Controllers



✦ dollar sign most likely something Angular defines for us (Angular specific variables). Sometimes called “view model.”

✦ our current model is the portfolios variable



✦

Adding a Controller

✦ clean up CSS by moving CSS to it and add bootstrap ‘panel-primary’ class in the div

```
<div class="panel panel-primary" ng-repeat="portfolio in...
```

✦ use angular 1.2.x for this to work

✦ in <script> tags at bottom of page before </body>

```
<script>
function PortfolioListController( ) {
}
</script>
```

✦ remove the ng-init and add the object to the controller. NB: remove the ng-init from the <body> element

```
function PortfolioListController () {
var portfolios = [
    {name: 'vessel1309', title: 'Call of Booty', date: ... },
    {name: 'innocents1404', title: 'The Sack of the ... },
    {name: 'firstmate1210', title: 'Pipe and First Mate'... }
    ];
}
</script>
```

Tell the View About the Controller

```
<body ng-controller="PortfolioListController">
```

✦ glue the model and view together. $scope tells the view where to get the data and then populate the view for the data via data binding.

```
function PortfolioListController ( $scope ) {
```

✦ edit

```
$scope.portfolios = [
    {name: 'vessel1309', title: 'Call of Booty', date:},...
```

✦ This separates the view from the model.

✦ Test the page in the browser. Note: no need to change anything in the interface.

Note - If necessary clear the lone paragraph with class cl

```
.cl { clear:both;}
```

Dependency Injection

```
function Class1 () {
   var user = new User();
}
function Class2 (user) {

}
```

The second example passes info into the function.

Dependency injection is simply giving a function an object (passing it an object) rather than creating an object inside a function.

Instead of creating an object for yourself, the managing framework declares the object. Example of framework for dependency injection (UD - 10)

Create the following and view it in the console.

```
var Person = function(firstname, lastname) {

    this.firstname = firstname;
    this.lastname = lastname;

}

function logPerson()
{
    var john = new Person('John', 'Doe');
    console.log(john);
}

logPerson();
```

✦ the function logPerson() is dependent on the variable John. This leads to poor and difficult to maintain code.

```
var Person = function(firstname, lastname) {

    this.firstname = firstname;
    this.lastname = lastname;

}

function logPerson(person)
{
    console.log(person);
}

var john = new Person('John', 'Doe');
logPerson(john);
```

Here the function is not dependent and the outside variable is “injected”. Person is created outside and passed to the function. Thus it can come from anywhere (i.e. a database).

Here all you care about Modular, someone else can own/create it.

Breakout Time

```
<!DOCTYPE html>
<html ng-app="myApp">
    <head>
        <title>AngularJS</title>
        <link rel="stylesheet" href="//
netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
/>
        <!-- load angular via CDN -->
        <script src="//code.angularjs.org/1.3.0-rc.1/
angular.min.js"></script>
        <script src="app.js"></script>
    </head>
```

```
    <body>
        <div class="container">
            <div ng-controller="mainController">
                <h1>Hello world!</h1>
            </div>
     </div>
    </body>
</html>
```

✦ app.js

```
var myApp = angular.module('myApp', []);
myApp.controller('mainController', function() {

});
```

```
1  var myApp = angular.module('myApp', []);
2
3  myApp.controller('mainController', function($scope) {
4
5      $scope.name = 'Jane Doe';
6      $scope.occupation = 'Coder';
7
8      $scope.getname = function() {
9          return 'John Doe';
10     }
11
12     console.log($scope);
13
14 });
```

Controllers / Forms to Add and Modify Data

Adding a Method to the controller

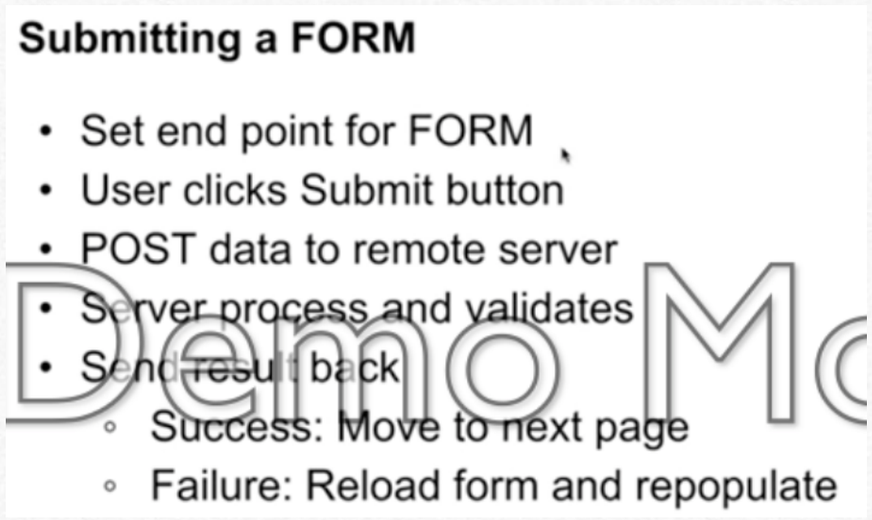
```
$scope.addPortfolio = function (new_portfolio) {
    $scope.portfolios.push(new_portfolio);
};
```

✦ push the new object onto the end of our array

✦ the ng-repeat will automatically update the view (data binding)

Forms in Angular

✦ usual method



✦ using the ng-repeat block as a template

✦ In preparation

```
<div class="panel-heading">
  {{ portfolio.name }} <span class="portfolio-
date">{{ portfolio.date }}</span>
</div>
```

and

```
.portfolio-date {
  display: inline-block;
  float:right;
}
```

✦ duplicate existing panel and make changes to add input fields

✦ in the panel-heading:

```
<div class="panel-heading">
  <input type="text" placeholder="title" size="20"
  ng-model="adding_portfolio.title" />
  <span class="portfolio-date">
    <input type="text" placeholder="yyyy/mm/dd" size="10" ng-
model="adding_portfolio.date" />
  </span>
</div>
```

✦ in the panel body

```
<div class="panel-body">
<textarea placeholder="description" rows="4" style="width:
100%;" ng-model="adding_portfolio.description"></textarea>

<input type="text" placeholder="short name" ng-
model="adding_portfolio.name" />

  <button type="button" class="btn btn-success">Add
Portfolio</button>
</div>
```

✦ add CSS for new elements as needed

```
input[type=text], textarea {
  padding: 3px;
  border-radius: 4px;
  border: 1px solid rgba(0,0,0,.2);
}
```

Invoking the addPortfolio method

✦ use the ng-click directive

```
<button ng-click="addPortfolio(adding_portfolio)"
type="button" class="btn btn-success">Add Portfolio</button>
```

✦ Test the page in a browser - note the ng-repeat error when adding a second portfolio. Its like the browser says “I’ve already seen this object in my array - what are you doing?”

✦ Also note the form isn’t cleared.

```
$scope.addPortfolio = function (new_portfolio) {
  $scope.portfolios.push(new_portfolio);
  $scope.adding_portfolio = {};
};
```

✦ Clear the adding_portfolio object, add as first line inside controller to make the process more explicit.

```
$scope.adding_portfolio = {};
```

HTML5 Forms

HTML 5 Validation

The fieldset element functions as a structural container for different sections within a form element.

For more information of HTML 5 forms please see: http://diveintohtml5.info/index.html

The label element attaches descriptive information to form elements like input fields, radiobuttons, textareas.

✦ HTML (add to start.html)

```
<form action="#" method="post">
<fieldset>
<legend>HTML Form</legend>

<label for="name">Name:</label>
<input type="text" name="name" required placeholder="Name" />
</fieldset>
</form>
```

✦ Other data types

```
<label for="email">Email:</label>
<input type="email" name="email" required
placeholder="email@example.com" />

<label for="website">Website:</label>
<input type="url" name="website" required placeholder="http://
www.example.com" />

<label for="number">Number:</label>
<input type="number" name="number" min="0" max="10" step="2"
required placeholder="Even num < 10">

<label for="range">Range:</label>
<input type="range" name="range" min="0" max="10" step="2" />

<label for="date">Date</label>
<input type="date" name="date" />

<label for="message">Message:</label>
<textarea name="message" required></textarea>

<input type="submit" value="Send Message" />
```

✦ In page CSS

```
body {
   font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
}
form {
   width:320px;
   margin: 20px auto;
}
ol {
   list-style: none;
   padding: 0;
}
*:focus{
```

```
  outline:none;
}
fieldset {
  border:none;
}
legend {
  font-size:24px;
  margin-bottom:20px;
}
input, textarea {
  border:1px solid #ccc;
  font-size:20px;
  min-width:300px;
  min-height:30px;
  display:block;
  margin-bottom:16px;
  margin-top:8px;
  border-radius:5px;
  transition: all 0.5s ease-in-out;
}
textarea {
  min-height:200px;
}
input:focus, textarea:focus {
  box-shadow:0 0 25px #ccc;
  transform: scale(1.05);
}
input:not(:focus), textarea:not(:focus) {
  opacity:0.5;
}
input:required, textarea:required {
  background:url("asterisk_orange.png") no-repeat 280px 7px;
}
input:valid, textarea:valid {
  background:url("tick.png") no-repeat 280px 5px;
}
input:focus:invalid, textarea:focus:invalid {
  background:url("cancel.png") no-repeat 280px 7px;
}
input[type=submit] {
  padding:10px;
  background:none;
  opacity:1.0;
}
-----------
```

Attribute Selectors Note

```
<h1 rel="xxxexternalxxx">Attribute Contains</h1>
h1[rel*="external"] { color: red; }

<h1 rel="external-link yep">Attribute Begins</h1>
h1[rel^="external"] { color: red; }

<h1 rel="friend external">Attribute Ends</h1>
h1[rel$="external"] { color: red; }

<h1 rel="friend external sandwich">Attribute Space Separated</
h1>
h1[rel~="external"] { color: red; }

<h1 rel="friend-external-sandwich">Attribute Dash Separated</
h1>
h1[rel|="external"] { color: red; }

<h1 rel="handsome" title="Important note">Multiple
Attributes</h1>
h1[rel="handsome"][title^="Important"] { color: red; }
-----------

input[type="number"], input[type="number"]:required,
input[type="number"]:valid, input[type="number"]:focus:invalid
{
```

```
  background-position:260px 7px;
}
```

Styling placeholder text -

http://css-tricks.com/snippets/css/style-placeholder-text/

```
::-webkit-input-placeholder {
   color: red;
}
:-moz-placeholder { /* Firefox 18- */
   color: red;
}
::-moz-placeholder {  /* Firefox 19+ */
   color: red;
}
```

Autofocus

```
<input type="text" name="name" required placeholder="Name"
autofocus />
```

Auto caps and correct

```
<input type="text" name="test1" autocapitalize="off" />
<input type="text" name="test2" autocorrect="off" />
```

Email vs. type="text"

```
<label for="email">E-mail</label>
<input name="email" id="email" type="email" required />
```

Input type = URL

```
<label for="website">Website</label>
<input name="website" id="website" type="url" />
```

Spinbox

```
if (!Modernizr.inputtypes.number) {
  // no native support for type=number fields
  // maybe try a JavaScript framework
}
```

novalidation

```
<form name="" action="" novalidate >
```

Angular Validation

novalidate

✦ use html5 form elements but not for validation

✦ Directives: https://docs.angularjs.org/api/ng/directive

✦ Filter: https://docs.angularjs.org/api/ng/filter

✦ Directive > Input https://docs.angularjs.org/api/ng/directive/input

✦ Overview forms handling from book samples

✦ check out Atlassian SourceTree

✦ Apple FileMerge

✦ NB - make sure in the opening controller

```
function PortfolioListController ($scope) {
    $scope.new_portfolio = {};
```

✦ NB in the add function portfolio after the push

```
$scope.adding_portfolio = {};
```

✦ NB - be sure to check/add the filter

```
<input type="text" placeholder="Search..." ng-
model="searchFor" size="20" ... />
```

```
  <div class="panel panel-primary" ng-repeat="portfolio in
portfolios  | filter: { title: searchFor } ">
```

Validation

```
<form name="" action="" novalidate >
```

✦ variables that tell us when things go wrong

✦ don't hang off the ng-model - doesn't want to pollute the model or the $scope object.

```
  <div class="panel panel-default">
<form name="add_portfolio_form" novalidate >
     <div class="panel-heading">
     <h3>Add new album</h3>
     <p>Valid? {{ add_portfolio_form.date.$valid }}</p>
     <p>Error? {{ add_portfolio_form.date.$error }}</p>
        <div style="float:right;">
<input ng-required="true" ng-minlength="10" name="date"
type="text" placeholder="yyyy/mm/dd" size="10" ng-
model="adding_portfolio.date" />
</div>
...
</div>
</form>
</div>
```

✦ test in browser

✦ add alert from bootstrap above form below panel opening, use ng-show

```
<div class="panel panel-default">
<div class="alert alert-danger"
//ng-show="add_portfolio_form.date.$error.minlength">
Date too short
</div>
```

```
  .alert-danger {
    display: block;
  }
  .ng-valid-minlength .alert-danger {
    display: none;
  }
```

✦ validation occurs when the user presses the Submit button

Make the CSS External and Clear Up the Display

```
...
textarea { width: 100%; margin-bottom: 12px; }
button {
    margin: 0 0 12px 12px;
}
.date {
    float: right;
}
...
.cl { clear: both }
.panel-heading a {
    color: #fff;
}
```

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

Adding Validation to the addPortfolio Method

✦ edit the alert box - REMOVE it from the form and place it above and outside

```
<div class="alert alert-danger" ng-show="add_portfolio_error">
{{add_portfolio_error}}
</div>
```

✦ empty string for the error in the scope

```
$scope.adding_portfolio = {};
```

```
        $scope.add_portfolio_error = "";
        $scope.portfolios = [
        {name: 'vessel1309', title: 'Call of Booty',
```

✦ in the addPortfolio function

```
$scope.addPortfolio = function (new_portfolio) {
  // validation goes here before the push
  $scope.portfolios.push(new_portfolio);
  $scope.adding_portfolio = {};
};
```

✦ validation ver 1

```
$scope.addPortfolio = function (new_portfolio) {

// if new_portfolio is not defined
if (typeof(new_portfolio) === 'undefined') {
    // then add an error message to $scope and exit
    $scope.add_portfolio_error = "The form is not properly
filled out";
    return false;
}

  if (!new_portfolio.date || new_portfolio.date.length < 10){
    $scope.add_portfolio_error = "You must provide a date in
format yyyy/mm/dd";
  }
  else
  {
    $scope.portfolios.push(new_portfolio);
    $scope.adding_portfolio = {};
  };
};
```

✦ validation ver 2

```
  $scope.addPortfolio = function( new_portfolio ){
    $scope.add_portfolio_error = "";
    // add validation below before the push
```

```
    if (!new_portfolio.title) {
      $scope.add_portfolio_error = "Missing title"
    }
    else if (!new_portfolio.date || new_portfolio.date.length
< 10){
      $scope.add_portfolio_error = "You must provide a date
in format yyyy/mm/dd";
    } else {
      $scope.portfolios.push( new_portfolio );
      $scope.adding_portfolio = {};
    };
  };
};
```

✦ validation ver 3

```
  $scope.addPortfolio = function( new_portfolio ){
    $scope.add_portfolio_error = "";
    // add validation below before the push
    if (!new_portfolio.title) {
      $scope.add_portfolio_error = "Missing title"
    }
    else if (!new_portfolio.date || new_portfolio.date.length
< 10){
      $scope.add_portfolio_error = "You must provide a date
in format yyyy/mm/dd";
    } else if (!new_portfolio.description){
      $scope.add_portfolio_error = "Missing description";
    } else if (!new_portfolio.name) {
      $scope.add_portfolio_error = "Missing name – six
characters";
    } else {
      $scope.portfolios.push( new_portfolio );
      $scope.adding_portfolio = {};
    };
  };
};
```

✦ Syntax Note - version w/o curly braces also works

```
  $scope.addPortfolio = function( new_portfolio ){
    $scope.add_portfolio_error = "";
    // add validation below before the push
    if (!new_portfolio.title)
      $scope.add_portfolio_error = "Missing title"
    else if (!new_portfolio.date || new_portfolio.date.length
< 10)
      $scope.add_portfolio_error = "You must provide a date
in format yyyy/mm/dd";
    else if (!new_portfolio.description)
      $scope.add_portfolio_error = "Missing description";
    else if (!new_portfolio.name)
      $scope.add_portfolio_error = "Missing name – six
characters";
    else {
      $scope.portfolios.push( new_portfolio );
      $scope.adding_portfolio = {};
    };
  };
};
```

✦ validation ver 4 - remove the error message after submit

```
else {
  $scope.portfolios.push(new_portfolio);
  $scope.adding_portfolio = {};
  $scope.add_portfolio_error = "";
}
```

Additional Validation 3.4 -

✦ date validation example

✦ remove

```
ng-required="true"
ng-minlength="10"

else if (!new_portfolio.date || !
is_valid_date(new_portfolio.date))
```

✦ the regex

```
/^[0-9]{4,4}/[0-9]{2,2}/[0-9]{2,2}$/
```

✦ the regex with escapes

```
function is_valid_date (the_date){
  //http://eloquentjavascript.net/09_regexp.html
  if (the_date.match(/^[0-9]{4,4}\/[0-9]{2,2}\/[0-9]{2,2}$/)){
    return true;
    return false;
  };
};
```

✦ with NaN and getTime()

```
function is_valid_date (the_date){
  //http://eloquentjavascript.net/09_regexp.html
  if (the_date.match(/^[0-9]{4,4}\/[0-9]{2,2}\/[0-9]{2,2}$/)){
    var d = new Date(the_date);
    return !(isNaN(d.getTime()));
  } else {
    return false;
  };
};
```

✦ see also phone number example

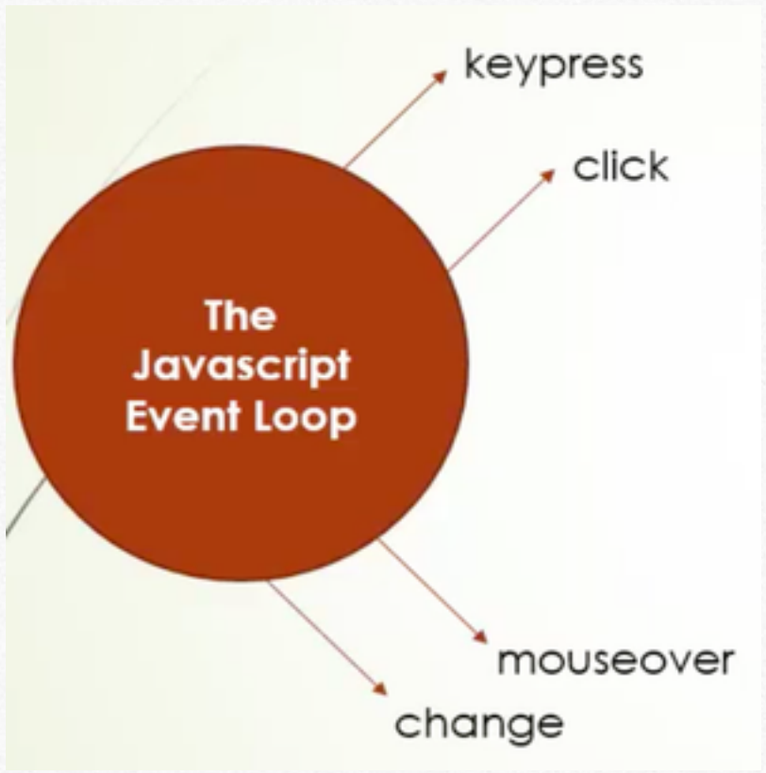
HTTP

✦ see also node example with npm

Watchers and the Digest Loop

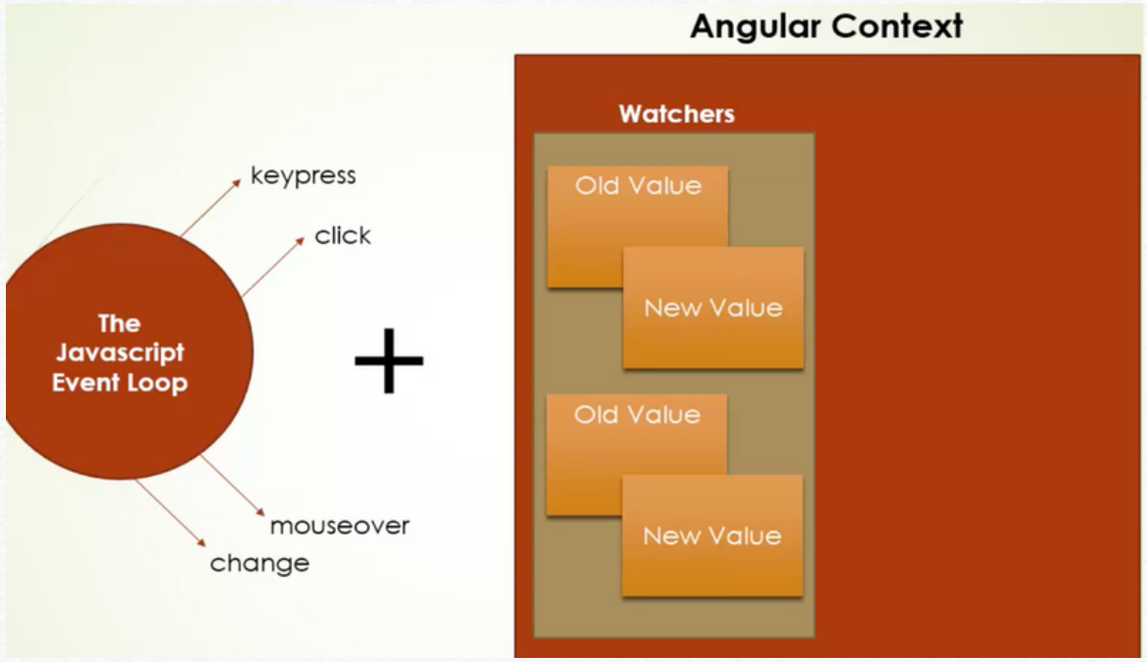
The JS Event Loop

Native to JavaScript



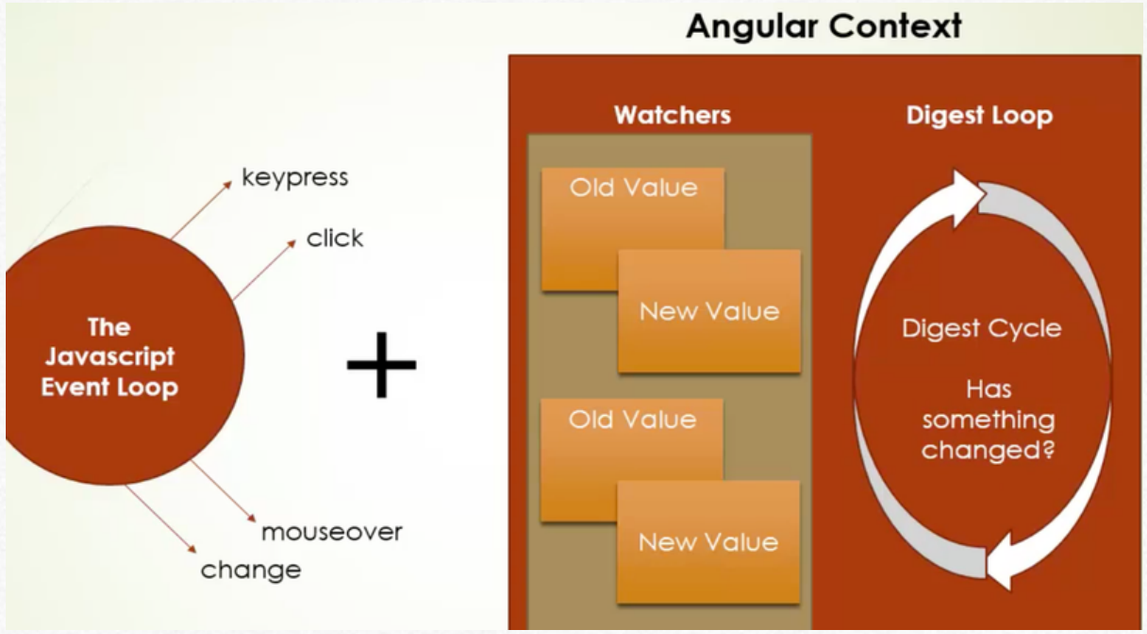
AngularJS extends the event loop in order to create and control the binding between the model and the view.

Angular automatically adds a “watcher” to the watchlist when you bind an element via $scope. It tracks the new and old value and whenever there is a native event fired it looks to see if any of its watchers are effected.



The part of Angular that checks for changes is inside the Digest Loop. This is Angular’s own event loop.

The digest loops looks to the watchers to see if anything has changed and, if so, then updates the value everywhere it is connected to - in both the DOM and the code.



c-add-watch demonstrates the digest cycle

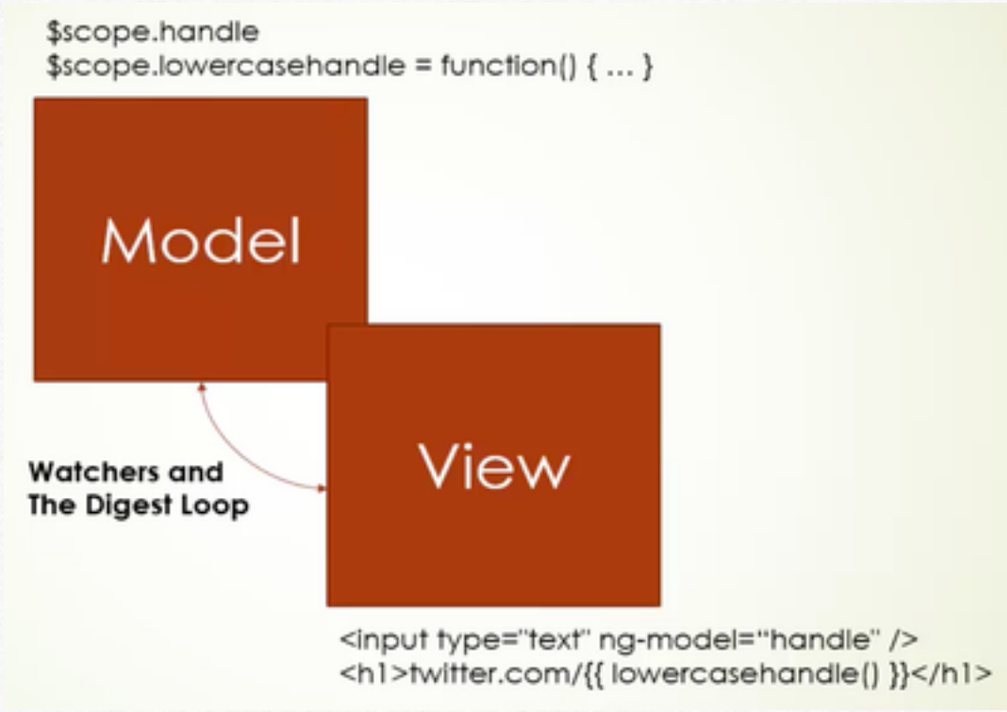
d-timeout-outside-angular sets up a new thread outside the digest cycle. It changed the scope but Angular didn't check for it. It didn't enter the digest loop.

e-apply overcomes this by adding it to the Digest loop.

Sometimes you want to do this i.e. when you are using another library like jQuery.

Caveat - you need to buy in all the way to Angular but you do not have to deal with all the plumbing code. You can focus on building the software.

The digest loop and watchers are what glue the model and the view together.



Includes and Controllers

Multiple Controllers

For currently logged in user. Make sure the current controller is in a div outside the controller

```
<div ng-controller="UserController">
  <p>Logged in as {{ user.name }}</p>
</div>
```

✦ outside the other controller

```
portfolioApp.controller('UserController', function( $scope ) {
    $scope.user = { name: "Daniel" };
});
```

ng-include

✦ create the directory app > partials

✦ ***note the quoted string***

```
<div style="float: right" ng-include="'app/partials/
user_partial.html'"></div>
```

✦ save this to the partials folder as user_partial.html

```
<div ng-controller="UserController">
  <p>Logged in as {{ user.name }}</p>
</div>
```

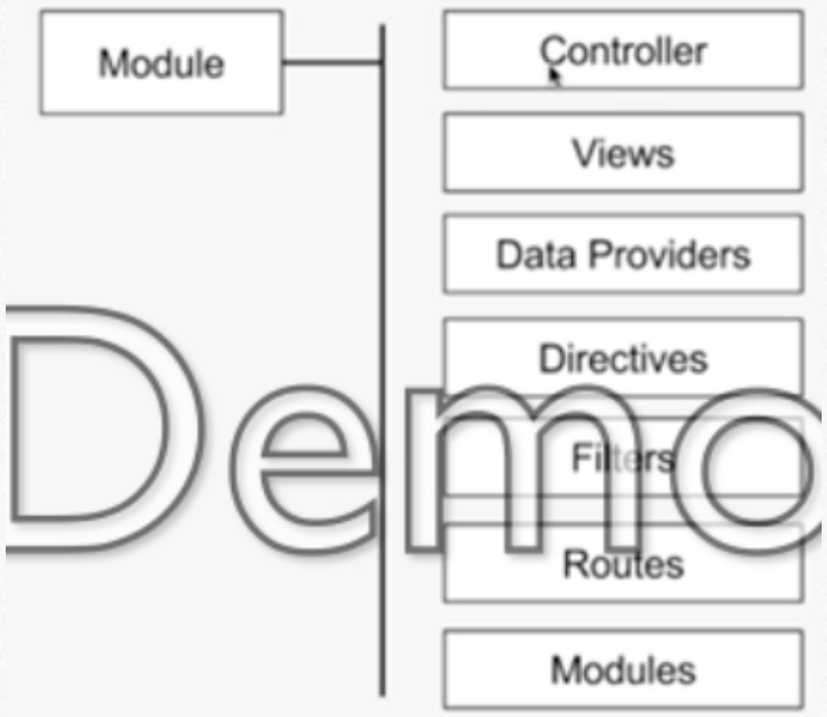
✦ test in browser

Homework: move the left column in the pirate portfolio into an include

Modules and Routes

Breaking Things Apart with Modules and Routing

✦ Module - a container for other parts of our application. Can contain many different elements.



JavaScript in Body

✦ module exists

```
<script type="text/javascript">
var portfolioApp = angular.module("portfolioApp", []);
```

✦ PortfolioListController. Using the controller() function. Pass in the name of the function as the second argument. Can also pass in an array of controllers.

```
portfolioApp.controller("PortfolioListController",
PortfolioListController);
```

✦ Add ng-app=”portfolioApp” to the head of the document

✦ Test in browser

Routes - 4.2

✦ Add route js to head and inject it into the module (dependency injection). Routes are not part of the core angularJS download.

```
<script src="https://code.angularjs.org/X.X.X/angular-
route.js"></script>
```

✦ injection into the app module

```
var portfolioApp = angular.module("portfolioSharingApp",
[ "ngRoute" ]);
```

✦ add function on our module a config function to provide a list of things to do when the user hits a particular url in our application.

✦ Place this after the module declaration (not in $scope):

```
portfolioApp.config(function ($routeProvider) {

});
```

✦ add route with path, controller and template

```
portfolioApp.config(function ($routeProvider) {
```

```
   $routeProvider
   .when("/portfolios",  { controller:
"PortfolioListController", templateUrl: "app/partials/
portfolios.html" });
});
```

✦ NOTE - going to /portfolios will not work at this point - why?

✦ you can also tell it to do other things via chaining using when and an otherwise clause (sort of an if/else structure)

```
portfolioApp.config(function ($routeProvider) {
   $routeProvider
      .when("/portfolios",  { controller:
"PortfolioListController", templateUrl: "app/partials/
portfolio_list_partial.html" })
      .when("/",  { redirectTo: "/portfolios" })
      .otherwise({ redirectTo: "/404_page" });
   });
```

Homework?

```
.when("/404_page",  { controller: “Controller404”,
templateUrl: "app/partials/404-page.html" })
```

Partials 4.2

✦ take all the *album list controller stuff* and create a view in partials

`portfolios.html`

We no longer need the controller directive in the partial as we have it in the route.

✦ In app > partials remove controller

```
<div ng-controller="PortfolioListController">
```

✦ add ng-view

```
<body>
   <ng-view></ng-view>
```

Laying out the application 4.3

✦ Add **controllers folder** to app directory

✦ here’s the entire app.js:

```
var portfolioApp = angular.module("portfolioSharingApp",
["ngRoute"]);

portfolioApp.config(function ($routeProvider) {
   $routeProvider
   .when("/portfolios",  { controller:
"PortfolioListController", templateUrl: "app/partials/
portfolio_list_partial.html" })
   .when("/",  { redirectTo: "/portfolios" })
   .otherwise({ redirectTo: "/404_page" });
   });
```

✦ move the portfolioListController (the rest of the javascript below routing) into controllers as portfoliolistcontroller.js

✦ This step NOT REALLY NECESSARY?
wrap it in a self executing function

```
(function(){

})();
```

✦ add the portfoliolistcontroller.js files *after app.js*

```
<script type="text/javascript" src="app/app.js"></script>
<script type="text/javascript" src="app/controllers/
portfoliolistcontroller.js"></script>
```

Adding a Portfolio View

✦ set up the titles as hyperlinks in partial (will need to change the color to white)

✦ In the ng-repeat portion of partial **portfolios.html** (ng-bind?)

```
<a href="index.html#/portfolio/{{portfolio.name}}">
  <span class="title" ng-bind="portfolio.title"></span>
</a>
  <span class="date" ng-bind="portfolio.date"></span>
...
```

✦ don't forget to add name attribute into the add portfolio form

```
<input name="name" type="text" placeholder="name"
ng-required="true" ng-model="adding_portfolio.name" />
???
<p class="error" ng-show="add_portfolio.name.$invalid &&
add_portfolio.name.$touched">Must enter a name.</p>
```

✦ set up the route in app.js using a **:portfolio_name parameter**. This is a feature built in to routes

```
.when("/portfolio/:portfolio_name",  {
   controller: "PortfolioViewController",
   templateUrl: "app/partials/portfolio.html"  })
.when("/",  { redirectTo: "/portfolios" })
```

✦

Portfolio View Partial

✦ create a new partial in partials folder, referencing the name in the routes "portfolio.html"

✦ create bootstrap div and ng-show for error text

```
<div class="alert alert-danger" ng-show="load_error_text">
  {{page_loading_error}}
</div>
```

✦ add bootstrap thumbnail class with temporary CSS with ng-repeat

```
<div class="photo thumbnail" ng-repeat="portfolio in
portfolios" style="width: 350px; float: left; margin-right:
10px">

</div>
```

✦ add image (note portfolio_name is from the controller as is $scope.portfolios) and description

```
<div class="portfolio thumbnail" ng-repeat="portfolio in
portfolios" style="width: 350px; float: left; margin-right:
10px">
<img src="/media/{{portfolio_name}}/thumb/
{{portfolio.filename}}" />
<p> {{ portfolio.description }} </p>
</div>
```

✦ add header

```
<h3>{{ portfolio_name }}</h3>
```

The Controller

✦ check app.js to ensure we are pointing to the new partial temple and...

✦ Create a new controller file as - portfolioviewcontroller.js file in the controllers directory

✦ create the controller function and register the controller with the portfolioApp

```
portfolioApp.controller("PortfolioViewController",
function($scope){

});
```

✦ inject $routeParams and associate the scope's portfolio name with the params name

```
portfolioApp.controller("PortfolioViewController",
function($scope, $routeParams){
   $scope.portfolio_name = $routeParams.portfolio_name;
});
```

✦ add the images - an object where the keys are the names of the objects and the arrays contain the images

✦ *you can pick this up from the other folder: portfolios.js*

```
portfolioApp.controller("PortfolioViewerController",
function($scope, $routeParams){
   scope.portfolio_name = $routeParams.portfolio_name;

var portfolios = {

'vessel1309': [
{ filename: "350.png",
date: "2013/09/05",
description: "I love this boat, so much booty." },
{ filename: "350.png",
date: "2013/09/06",
description: "We had a wonderful time on her."}],

'innocents1404': [
{ filename: "350.png",
date: "2014/04/14",
description: "So cold and so much sacking!" },
{ filename: "350.png",
date: "2014/04/15",
description: "The sails are so white here."}],

'firstmate1210': [
{ filename: "350.png",
date: "2012/10/01",
description: "Getting mah pipe on!"},
{ filename: "350.png",
date: "2012/10/02",
description: "FTW!!!11!one!1"}]
};
});
```

✦ note the photos are set up already and note the directory structure:
media > [key] > [full ll thumb] > 350.png

```
portfolioApp.controller("PortfolioViewerController",
function($scope, $routeParams){
   $scope.portfolio_name = $routeParams.portfolio_name;
   $scope.load_error_text = "";
var portfolios = {
    ...
};
if (portfolios[$scope.portfolio_name]) {
   $scope.portfolios = portfolios[$scope.portfolio_name];
   console.log($scope.portfolios);
} else {
   $scope.load_error_text = "Aaargh! I can't find the darn
portfolio";
};
});
```

✦ Add reference to the new js to the index.html file

```
<script type="text/javascript" src="app/controllers/
portfolioviewcontroller.js"></script>
```

Pirate Data Aaaargh!

Get the portfolio data to FB

✦ add the constant to the student's FB url and dependancies to **app.js**

```
var portfolioApp = angular.module('portfolioApp',
['ngRoute','firebase']).constant('FIREBASE_URL', 'https://
pirate-portfolio.firebaseio.com/');
```

✦ add the firebase url's to **index.html**

```
<script src="https://cdn.firebase.com/js/client/2.2.2/
firebase.js"></script>
<script src="https://cdn.firebase.com/libs/angularfire/1.0.0/
angularfire.min.js"></script>
```

✦ structure the data (do only vessel1309 for the moment) in a new document leaving only $scope.portfolios = []; in **portfolioListController.js**

```
{
  name: 'vessel1309',
  title: 'Call of Booty',
  date: '2013-09-01',
  description: 'Arrrgh mateys! This be the finest vessel
sailing these seas that yer eyes ever laid sight on. Arrrgh.',
  imageurl: 'images/drunkenPirate_thumb3.png' }
```

✦ https://www.firebase.com/docs/web/api/firebase/set.html

```
portfolioApp.controller("PortfolioListController",
function($scope, $firebaseAuth, $firebaseArray, FIREBASE_URL){

  $scope.portfolios = [];

  var ref = new Firebase(FIREBASE_URL);
  var postRef = ref.child('portfolios');

  postRef.set({
    'vessel1309': {
      title: 'Call of Booty',
      date: '2013-09-01',
      description: 'Argh mateys! This be the finest vessel
sailing these seas that yer eyes ever laid sight on. Arrrgh.',
      imageurl: 'images/drunkenPirate_thumb3.png'
    }
  });

  // $scope.addPortfolio = function( new_portfolio ) {
  //   $scope.portfolios.push( new_portfolio );
  //   $scope.add_portfolio = {};
  // };

});
```

✦ then add the other portfolios

```
postRef.set({
  'vessel1309': {
    title: 'Call of Booty',
    date: '2013-09-01',
    description: 'Argh mateys! This be the finest vessel
sailing these seas that yer eyes ever laid sight on. Arrrgh.',
    imageurl: 'images/drunkenPirate_thumb3.png'
  },
```

```
  'innocents1404': {
    title: 'The Sack of the Innocents',
    date: '2014-04-15',
    description: 'This be a detail of the sacking of the
innocents where I played an important role as business analyst
and in implementation.',
    imageurl: 'images/drunkenPirate_thumb2.png'
  },

  'firstmate1210': {
    title: 'Pipe and First Mate',
    date: '2012-10-01',
    description: 'After a hard day of lootin\' and shootin\' I
like to relax with my best mates and tally up the days
booty.',
    imageurl: 'images/drunkenPirate_thumb1.png'
  }
});
```

✦ set the value of $scope.portfolios

```
$scope.portfolios = $firebaseArray(postRef);
```

✦ some useful items

```
console.log(postRef.key());

postRef.on("value", function(snapshot) {
  console.log(snapshot.val());
}, function (errorObject) {
  console.log("The read failed: " + errorObject.code);
});
```

✦ NB - the second page is broken - **routeParams**

✦ add name: xxxx to the FB data

```
postRef.set({
  'vessel1309': {
    name: 'vessel1309',
    title: 'Call of Booty',
    date: '2013-09-01',
    description: 'Argh mateys! This be the finest vessel
sailing these seas that yer eyes ever laid sight on. Arrrgh.',
    imageurl: 'images/drunkenPirate_thumb3.png'
  }
```

Get the image data to FB

✦ in **portfolioviewcontroller.js**

```
var pushImg = new Firebase(FIREBASE_URL  + '/images');
pushImg.set({
  'vessel1309': [
  { filename: "350.png",
  date: "2013/09/05",
  description: "I love this boat, so much booty." },
  { filename: "350.png",
  date: "2013/09/06",
  description: "We had a wonderful time on her."}],

  'innocents1404': [
  { filename: "350.png",
  date: "2014/04/14",
  description: "So cold and so much sacking!" },
  { filename: "350.png",
  date: "2014/04/15",
  description: "The sails are so white here."}],

  'firstmate1210': [
  { filename: "350.png",
  date: "2012/10/01",
  description: "Getting mah pipe on!"},
  { filename: "350.png",
  date: "2012/10/02",
  description: "FTW!!!11!one!1"}]
```

```
});
```

✦ then

```
portfolioApp.controller("PortfolioViewerController",
function($scope, $routeParams, FIREBASE_URL, $firebaseArray,
$firebaseObject){

 $scope.portfolio_name = $routeParams.portfolio_name;
 $scope.load_error_text = "";

var portfolio_name = $routeParams.portfolio_name;

var imgRef = new Firebase(FIREBASE_URL  + '/images/'+
portfolio_name);

$scope.portfolios = $firebaseArray(imgRef);
```

✦ to do: re-implement the error checking e.g.

```
    if ($scope.portfolios[$scope.portfolio_name]) {
      $scope.portfolios = $firebaseArray(imgRef);
    } else {
      $scope.load_error_text = "Aaaargh! I can't find that
darn portfolio matey!";
    }
```

✦ implement a back button

Get the form working

✦ using push in **portfolioListController.js**

```
$scope.addPortfolio = function( new_portfolio ) {
  var pushRef = new Firebase(FIREBASE_URL + '/portfolios');
  pushRef.push({
    name: new_portfolio.name,
    creationdate: Firebase.ServerValue.TIMESTAMP,
    date: new_portfolio.date,
    title: new_portfolio.title,
    description: new_portfolio.description,
    image: new_portfolio.image
  });
  $scope.add_portfolio = {};
};
```

Uploading image using filepicker.com

✦ in index.html

```
<script type="text/javascript" src="//api.filepicker.io/v1/
filepicker.js"></script>

<!-- AgTJpb5i3RlKpbtqVuZSAz  -->
```

✦ in the form

```
<input type="file" id="file-upload" ng-click="addImage()">
```

✦ in the controller

```
$scope.addImage = function( new_image ){
// https://www.filepicker.com/documentation/file_ingestion/
javascript_api/pick
filepicker.setKey("AgTJpb5i3RlKpbtqVuZSAz");

  filepicker.pick(
  {
    mimetypes: ['image/*', 'text/plain'],
    container: 'window',
    services:['COMPUTER'],
  },
  function(Blob){
    console.log(JSON.stringify(Blob));
  },
  function(FPError){
    console.log(FPError.toString());
```

```
  }
)};
```

✦ get the console on success into a variable and include it in our data push

```
var imageupped;

$scope.addImage = function( new_image ){
// https://www.filepicker.com/documentation/file_ingestion/javascript_api/pick
filepicker.setKey("AgTJpb5i3RlKpbtqVuZSAz");
  filepicker.pick(
  {
    mimetypes: ['image/*', 'text/plain'],
    container: 'window',
    services:['COMPUTER'],
  },
  function(Blob){
    console.log(JSON.stringify(Blob));
    imageupped = Blob.url;
  },
  function(FPError){
    console.log(FPError.toString());
  }
)};
```

✦ form

```
$scope.addPortfolio = function( new_portfolio ) {
    console.log(imageupped);
  var pushRef = new Firebase(FIREBASE_URL + '/portfolios/');
  var imagePath ="/media/"+ new_portfolio.name + "/thumb/";
  console.log('test ' + imageupped);
  pushRef.push({
    name: new_portfolio.name,
    creationdate: Firebase.ServerValue.TIMESTAMP,
    date: new_portfolio.date,
    title: new_portfolio.title,
    description: new_portfolio.description,
    imageurl: imageupped
});
  $scope.add_portfolio = {};
};
```

Date Object

Adding links on include file

```
  <nav class="menus" ng-controller="PortfolioListController">
    <h3 class="work">Work</h3>
    <ul>
      <li ng-repeat="portfolio in portfolios">
<a href="index.html#/portfolio/{{portfolio.name}}">
{{ portfolio.title }}
      </a></li>
    </ul>
```

Create a simple data extraction -

- ✦ login to filepicker and firebase
- ✦ make changes to CONSTANT and check FB and FP for developer key.
- ✦ If no new portfolio has been added, add one using the image in other folder
- ✦ note the form and FB entry: creationDate
- ✦ add to portfolios.html

```
<span class="date" ng-bind="portfolio.date"></span>
<span class="date" ng-bind="creationDate"></span>
```

?

- ✦ getting information out of a data snapshot (time in milliseconds since unix epoch).
- ✦
- ✦ GOAL - the date should look like this **2 May 2015**
- ✦ From:

```
postRef.on('value', function(snapshot){
  console.log(snapshot.val());
}, function (errorObject) {
  console.log("The read operation failed: " +
errorObject.code);
});
```

- ✦ to

```
var portSnapshot;
  postRef.once('value', function(dataSnapshot) {
     portSnapshot = dataSnapshot;
     var portDate = portSnapshot.child('-
JoMm3UOg9ZBqlr4yuYl').child('creationDate');
     var firstItem = portDate.val();
     console.log(firstItem);
   }, function (errorObject) {
    console.log("The read operation failed: " +
errorObject.code);
  });
```

- ✦ this gets the long unformatted date into the view - html partial

```
<span class="date" ng-bind="creationDate"></span>
```

- ✦ in controller

```
var date;
postRef.once('value', function(dataSnapshot) {
  portSnapshot = dataSnapshot;
```

```
  var portDate = portSnapshot.child('-
JoMm3UOg9ZBqlr4yuYl').child('creationDate');
  var portdateVal = portDate.val();
  date = new Date(portdateVal);
  $scope.creationDate = date;
});
```

Famous Pirates
Great pirate, great looter.

Sat May 02 2015
23:18:31 GMT-0400
(EDT)

✦ create a service for date formatting - note the [ ] brackets

portService.js

```
portfolioApp.service("PortService", [function($scope) {
  console.log('made it inside port service');
}]);
```

✦ register the service in index.html and the controller portfolioListController

✦ explore the date object inside the service.js

```
var monthNames = [
    "January", "February", "March",
    "April", "May", "June", "July",
    "August", "September", "October",
    "November", "December"
];

var date = new Date();
var day = date.getDate();
var monthIndex = date.getMonth();
var year = date.getFullYear();

console.log('This is the date result: ' + day,
monthNames[monthIndex], year);
```

✦ use the dataviz template (authentication.js / registration.js) to build return obj in **portService.js**

```
portfolioApp.service("PortService", [function($scope) {
  console.log('made it inside port service');

  var myObject = {
    formatDate: function(serverdate) {

      var monthNames = [
      "January", "February", "March",
      "April", "May", "June", "July",
      "August", "September", "October",
      "November", "December"
      ];

      var date = new Date();
      var day = date.getDate();
      var monthIndex = date.getMonth();
      var year = date.getFullYear();

      console.log('This is the date result: ' + day,
monthNames[monthIndex], year);
    }};
    return myObject
  }]);
```

✦ this inserts the date of the portfolio instead of today's date

```
var monthNames = [
"January", "February", "March",
"April", "May", "June", "July",
"August", "September", "October",
"November", "December"
];

var date = new Date(serverdate);
var day = date.getDate();
```

✦ **FYI -** At this point I was able to change this to a factory

```
portfolioApp.factory("PortService", [function($scope) {
```

http://stackoverflow.com/questions/15666048/service-vs-provider-vs-factory

✦ in service/factory

```
    var date = new Date(serverdate);
    var day = date.getDate();
    var monthIndex = date.getMonth();
    var year = date.getFullYear();
    dateResult = (day + ' ' + monthNames[monthIndex] + ' ' +
year);
    return dateResult;
  }};
  return myObject
}]);
```

✦ in controller

```
var portdateVal = portDate.val();
date = new Date(portdateVal);
$scope.creationDate = PortService.formatDate(date);
// $scope.creationDate = date;
console.log(dataSnapshot.val());
```

NOTES

```
  var myObject = {
    formatDate: function(serverdate) {
      var d = new Date(serverdate);
      console.log(d);
    },
    }; // myobj
    return myObject
}]);
```

✦ add to portfolioListController.js

```
var portDate = portSnapshot.child('-
JoMm3UOg9ZBqlr4yuYl').child('creationDate');
var firstItem = portDate.val();
PortService.formatDate(firstItem);
```

✦ Expanding on the obj in **portService.js**

```
var myObject = {
formatDate: function(serverdate) {
```

```
    var d = serverdate.getUTCDate().toString(),
        m = (serverdate.getUTCMonth() + 1).toString(),
        y = serverdate.getUTCFullYear().toString(),
        formatted = '';
    console.log(y); // changing this as we go
```

✦ meanwhile in **portfolioListController.js**

```
      var portSnapshot;
      postRef.once('value', function(dataSnapshot) {
       portSnapshot = dataSnapshot;
       var portDate = portSnapshot.child('-
JoMm3UOg9ZBqlr4yuYl').child('creationDate');
       var portdateVal = portDate.val();
       var date = new Date(portdateVal);
       PortService.formatDate(date);
```

✦ add the if statements - **portfolioListController.js**

```
       formatted = '';
       if (d.length === 1) {
           d = '0' + d;
       }
       if (m.length === 1) {
           m = '0' + m;
       }
       console.log(m);
```

✦ complete portService

```
var myObject = {
formatDate: function(serverdate) {
    var d = serverdate.getUTCDate().toString(),
        m = (serverdate.getUTCMonth() + 1).toString(),
        y = serverdate.getUTCFullYear().toString(),
        formatted = '';
        if (d.length === 1) {
            d = '0' + d;
        }
        if (m.length === 1) {
```

```
            m = '0' + m;
        }
        formatted = d + '-' + m + '-' + y;
        return formatted;
    }}; // myobj
    return myObject
}]);
```

Pirate Pickup

Picking up from (near) where we left off

Adding routing and controllers

✦ do a review of where it stands.

✦ NB have already broken out left-col in **index.html**

```
<aside class="sidebar" ng-include="'includes/left-
col.html'"></aside>
```

✦ breakout and updating to angular 1.3x (in app/js/) - **index.html**

```
<!DOCTYPE html>
<html ng-app="portfolioApp">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="initial-scale=1.0">

  <title>Drunken Pirate Portfolio</title>
  <link rel="stylesheet" href="css/style.css">
  <script src="app/js/angular.min.js"></script>
  <script src="app/js/angular-route.min.js"></script>
  <script src="app/app.js"></script>
  <script src="app/controllers/portfolioListController.js">
</script>
</head>

<body>
  <main class="wrap group">
    <aside class="sidebar" ng-include="'includes/left-
col.html'"></aside>
    <div ng-view></div>
  </main>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1/
jquery.min.js"></script>
</body>
</html>
```

✦ create **partials/portfolios.html**

```
<div class="content">
   ...
</div>
```

✦ in app/**app.js - (NEW)**

```
var portfolioApp = angular.module("portfolioApp",["ngRoute"]);

var PortfolioListController =
angular.module("PortfolioListController", []);

portfolioApp.config(["$routeProvider",
function($routeProvider) {

  $routeProvider.
  when("/", {
    templateUrl: "partials/portfolios.html",
    controller: "PortfolioListController"
  }).
  otherwise({
    redirectTo: "/404"
  });

}])
```

✦ remove references to controller from index.html

```
<body ng-controller="PortfolioListController">
  <main class="wrap group">
```

✦ copy controller from script block in **controllers/ portfolioListController.js** and make changes

```
portfolioApp.controller("PortfolioListController",
function($scope){
$scope.adding_portfolio = {};
$scope.portfolios = [
    ...
```

```
    ];
    $scope.addPortfolio = function( new_portfolio ) {
      $scope.portfolios.push( new_portfolio );
      $scope.adding_portfolio = {};
    };
});
```

✦ test in browser

✦ test adding a portfolio

Routing

5

Made to coincide with simple routing examples. Should use files from book as examples as well.

Routing

Routing and animation

Develop the index

```
<!DOCTYPE html>
<html lang="en" ng-app="helloWorldApp">
<head>
    <title>Class Review</title>

    <script src="js/libs/angular.min.js"></script>
    <script src="js/libs/angular-route.min.js"></script>

    <script src="js/app.js"></script>
    <script src="js/controllers.js"></script>

</head>
<body>
    <div ng-view></div>
</body>
</html>
```

Develop the app.js file.

```
var helloWorldApp = angular.module('helloWorldApp',
    [
    'ngRoute',
    'helloWorldControllers'
    ]);

helloWorldApp.config(['$routeProvider',
    function($routeProvider) {
        $routeProvider.
        when('/', {
            templateUrl: 'partials/main.html',
            controller: 'MainCtrl'
        }).when('/show', {
            templateUrl: 'partials/show.html',
            controller: 'ShowCtrl'
        }).otherwise({
            templateUrl: 'partials/404.html',
            controller: 'FourCtrl'
        });
    }]);
```

==========

Develop the controller

```
var helloWorldControllers =
angular.module('helloWorldControllers', []);

helloWorldControllers.controller('MainCtrl', ['$scope',
    function MainCtrl($scope) {
        $scope.pageClass = 'page-home';
        $scope.message = "Hello World";
    }]);

helloWorldControllers.controller('ShowCtrl', ['$scope',
    function ShowCtrl($scope) {
        $scope.pageClass = 'page-about';
        $scope.message = "Show The World";
    }]);
```

```
helloWorldControllers.controller('FourCtrl', ['$scope',
    function FourCtrl($scope) {
        $scope.pageClass = 'page-404';
        $scope.message = "404 - you are lost!";
    }]);
```

==========

Animation

✦ Ideas taken from - http://tympanus.net/codrops/2013/05/07/a-collection-of-page-transitions/

Page Class

✦ Add the page class directive and examine the HTML in inspector

```
<div class="page {{ pageClass }}" ng-view></div>
```

✦ Add the angular animate scripts

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.7/angular-animate.js"></script>
```

✦ Inject ngAnimate into the module

```
var helloWorldApp = angular.module('helloWorldApp', [
    'ngRoute',
    'ngAnimate',
    'helloWorldControllers'
]);
```

ngAnimate - adds and removes CSS classes to different Angular directives based on if they are entering or leaving the view. For example, when we load up a site, whatever is populated in ng-view gets a .ng-enter class.

ngAnimate Works On: ngRepeat, ngInclude, ngIf, ngSwitch, ngShow, ngHide, ngView, and ngClass

http://docs.angularjs.org/api/ngAnimate

✦ Examine animation.css

- make our pages be full width and full height

- positioned absolutely so that the pages can overlap each other as they enter and leave

- 6 different animations - each page will have their very own ng-enter and ng-leave animation

✦ Move it to the working directory and link it

```
<link rel="stylesheet" type="text/css" href="animation.css">
```

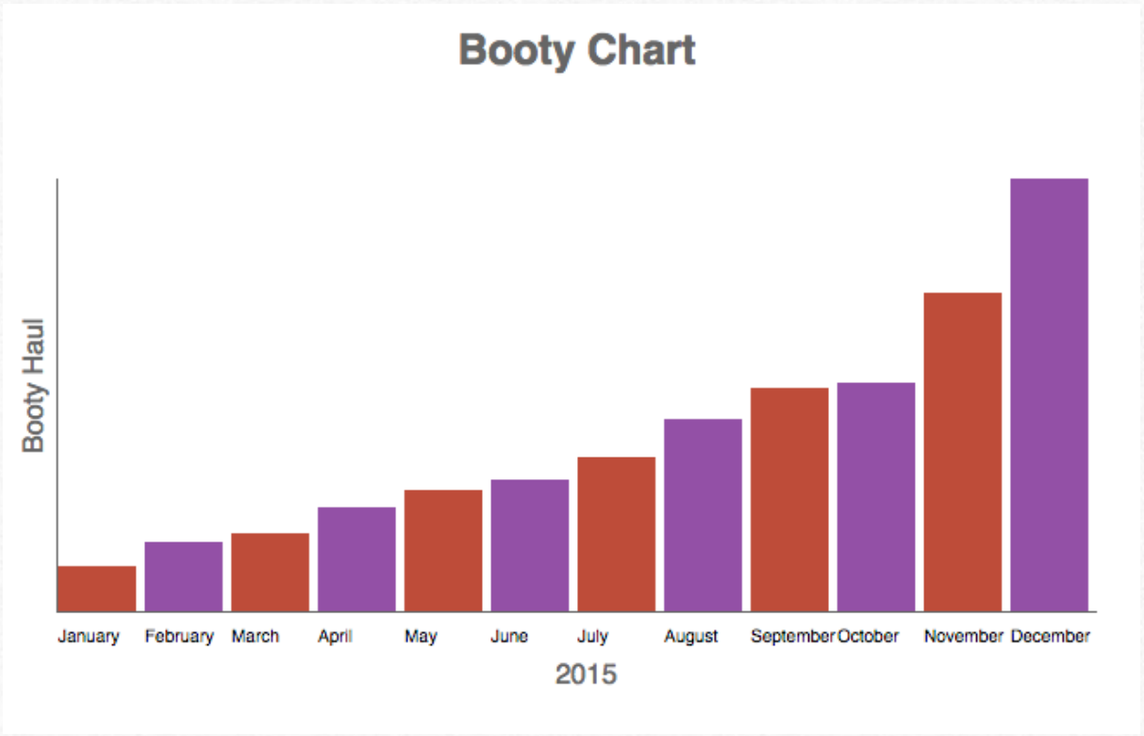
Dataviz and Firebase

6

Lorem ipsum dolor rutur amet. Integer id dui sed odio imperd feugiat et nec ipsum. Ut rutrum massa non ligula facilisis in ullamcorper purus dapibus.

Data Visualization with AngularJS

Simple Data Visualization



start.html

✦ ng-app directive on html tag

```
<html ng-app>
```

ng-include

add navigation bar in partials as navigation.html

```
<h2><a ng-href="#/">Booty Chart</a></h2>
<a ng-href="#/graph">graph</a>
<a ng-href="#/login">login</a>
<a ng-href="#/register">register</a>
```

✦ include it on the index page

```
<nav class="nav" ng-include="'partials/navigation.html'">
</nav>
```

✦ style it in scss file (start Koala)

```
nav * {
  display:inline-block;
  margin-right: 20px;
}
```

Create the Module

```
<html ng-app="graphApp">
```

✦ note error

✦ in blank app.js - namespace

```
var graphApp = angular.module('graphApp',[]);

var graphApp = angular.module('graphApp',['ngRoute']);

var graphApp = angular.module('graphApp',['ngRoute',
'appControllers']);
```

✦ note that app and controllers js files are already added in the index file

```
  <script src="app/app.js"></script>
  <script src="app/controllers/controllers.js"></script>
```

✦ define the controllers as another module in app.js

```
var appControllers = angular.module('appControllers', []);
```

✦ add routes

```
graphApp.config(['$routeProvider', function($routeProvider) {
    $routeProvider.
    when('/graph',{
        templateUrl: 'partials/graph.html'
    })
    .when('/login',{
        templateUrl: 'partials/login.html'
    })
    .when('/register',{
        templateUrl: 'partials/register.html'
    })
    .otherwise({
     redirectTo: '/login'
    });
}]);
```

✦ note error in browser

✦ prepare graph.html by cutting and pasting html into graph.html files

```
<div class="wrap">
  <div class="chart">
    <div class="y">
    </div>
    <div class="x">
    </div>
    <div class="bar">
    </div>
  </div>
```

✦ prepare login.html by creating login.html in partials

```
<h1>Login</h1>
<p>Log in to access the site</p>
```

✦ save as register.html and change the content

✦ add ng-view into index.html

```
<main class="wrap" ng-view>
</main>
```

✦ test routes in localhost

Implementing the controller

✦ add to controller.js file nesting the Data block

```
graphApp.controller('graphControllers', function($scope){
  --Data--
});
```

✦ add controller reference to the ***routing***

```
when('/graph',{
    templateUrl: 'partials/graph.html',
    controller: 'graphControllers'
})
```

✦ add options just above data block

```
// Options
$scope.width = 600;
$scope.height = 250;
$scope.yAxis = 'Booty Haul';
$scope.xAxis = '2015';
```

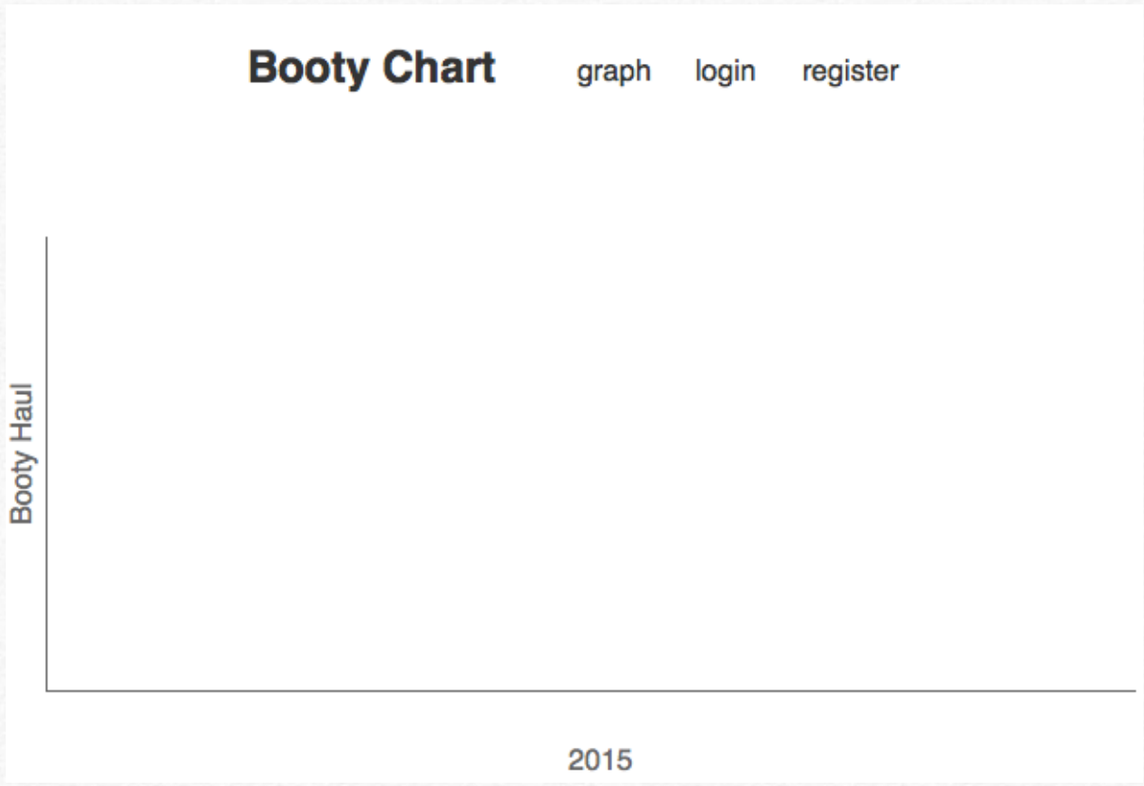
✦ edit graph.html - add inline CSS

```
<div class="chart" style="width:{{width}}px; height:{{height}}
px;">
...
<div class="y" style="width:{{height}}px;">{{yAxis}}</div>
<div class="x">{{xAxis}}</div>
```

'y' div uses the {{height}} variable for the width CSS property because in the CSS we rotate this counter clockwise by 90 degrees

```
.chart {
  border-left:1px solid #666;
```

```
  border-bottom:1px solid #666;
  margin:60px auto;
  position:relative;
}
.y {
  position:absolute;
  -webkit-transform:rotate(-90deg);
  -webkit-transform-origin: bottom left;
  bottom:0;
  padding:5px;
}
.x {
  position:absolute;
  top:110%;
  width:100%;
  padding:5px;
}
```



✦ add bar in data by editing the bar div (see explanations below)

```
<div class="bar" ng-repeat="bar in data"
style="
height:{{bar.value / max * height}}px;
width:{{width / data.length - 5}}px;
left:{{$index / data.length * width}}px;">
```

✦ **height** is the bar value, divided by the maximum (need to set up in our controller), multiplied by the total height of our chart

```
e.g. bar one:
      {
        label: 'January',
        value: 36
      }
  {{height}} = 250
  {{max}} = 342 (maximum height or the highest value in the
data)
  height :   36 / 342 * 250 = 26.315...
  if you wish to have a fluid chart you could multiply by 100
and use % as the unit rather than pixels
```

✦ **width** is the width of the chart divided by the number of entries, with 5px knocked off to create some spacing for our bars on the X axis

✦ **left** uses $index, an Angular variable that will start at 0 and increase for each subsequent bar. We divide the index by the total number of entries and multiply this by the full width of the chart. This places the first bar at 0 and then spaces the rest of the bars equally across the chart

✦ we need to set the max to the highest value in Data (after the data block)

```
$scope.max = 342;
```

✦ style bars (try to evolve this)

```
.bar  {
  background:#BE4C39;
  position:absolute;
  bottom:0;
}
.bar:nth-of-type(even) {
  background:#9351A6;
}
```

✦ Try evolving this para in the bar div (see the explanations of height, width and left)

```
<p style="color:#fff;">{{bar.value}}px</p>
<p style="color:#fff;">{{bar.value / max * height}}px</p>
```

✦ add and style the labels (they go into the ng-repeat region)

```
<p class="label" style="color:#000;">{{bar.label}}</p>

.label {
  position: absolute;
  bottom: -30px;
  font-size: 10px;
}
```

✦ optional

Find Maximum X & Y Axis Values - this is used to position the data as a percentage of the highest value

```
$scope.max = 0;
var arrLength = $scope.data.length;
console.log($scope.data.length);
```

✦ write a loop to cycle through our data to find the maximum value and set this as a variable

```
for (var i = 0; i < arrLength; i++) {
  // Find Maximum X Axis Value
  if ($scope.data[i].value > $scope.max)
    $scope.max = $scope.data[i].value;
}
```

✦ test by changing the highest value e.g.

```
    {
      label: 'December',
      value: 642
    }
```

Firebase

✦ review the instructions on firebase, add accounts

https://www.firebase.com/docs/web/libraries/angular/quickstart.html

✦ add the firebase links to index.html after the angular js script

```
<script src="https://cdn.firebase.com/js/client/2.2.2/
firebase.js"></script>
<script src="https://cdn.firebase.com/libs/angularfire/1.0.0/
angularfire.min.js"></script>
```

✦ inject firebase into app.js

```
var graphApp = angular.module('graphApp',['ngRoute',
'appControllers','firebase']);
```

✦ edit the controller

```
app.controller('graphController', function($scope,
$firebaseArray){

var ref = new Firebase('https://vizapp.firebaseio.com/data');
$scope.graphData = $firebaseArray(ref);
```

✦ change the reference in graph.html

```
<div class="bar" ng-repeat="bar in graphData" ...
```

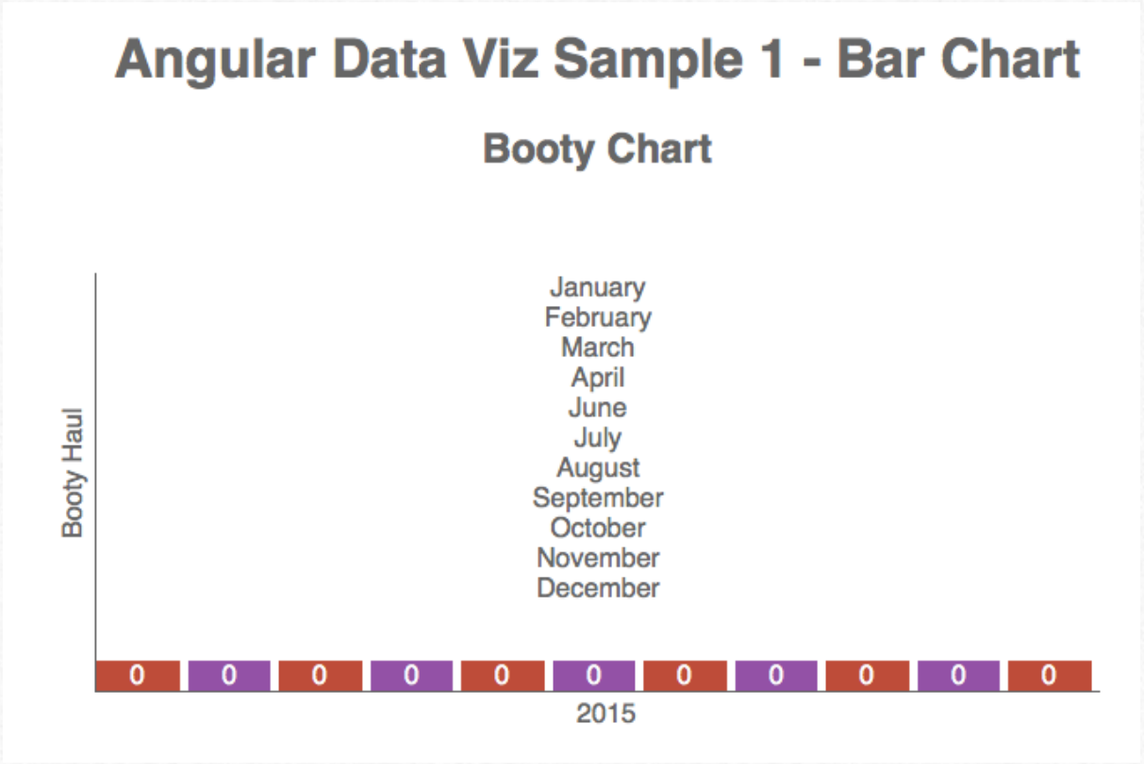
✦ edit the style values

```
<div class="bar" ng-repeat="bar in graphData"
  style="height:{{bar.value / max * height}}px;
```

```
  width:{{width / graphData.length - 5}}px;
  left:{{$index / graphData.length * width}}px;">
```

✦ may need to set the max value and remove the for loop (if implemented)

```
$scope.max = 342;
```



✦ N.B. The loop through array is not working to find the highest value

NB - Add a graph

✦ place this into the graph partial at bottom

```
<section>
  <h3>Add Graph</h3>
  <form name="myform"
  ng-submit="addGraph()"
  novalidate>
    <input type="text" name="graphname"
    placeholder="New graph name"
    ng-model="graphname" ng-required="true">

    <button type="submit" ng-disable="myform.$invalid">Add
    </button>
    </form>
<h4>Your Graphs</h4>
    <div ng-repeat="graph in graphs">
      <p>{{graph.name}}</p>
    </div>
</section>
```

Login and Register

✦ add this to the login.html partial

```
<section class="login">
  <form name="myform">
    <h1>Login</h1>
    <p>Log in to access the site</p>
    <input type="email" name="email" placeholder="Email"
autofocus>
    <input type="password" name="password"
placeholder="password">
    <button type="submit" class="btn">Login</button>
    <p>or <a href="#/register">Register</a></p>
  </form>
</section>
```

✦ add the ng-models - login.html

```
<section class="login">
  <form name="myform">
    <h1>Login</h1>
    <p>Log in to access the site</p>
```

```
     <input type="email" name="email" placeholder="Email"
autofocus ng-model=”user.email”>
     <input type="password" name="password"
placeholder="password" ng-model=”user.password”>
     <button type="submit" class="btn">Login</button>
     <p>or <a href="#/register">Register</a></p>
   </form>
</section>
```

✦ add validation - login.html

```
<section class="login">
   <form name="myform" novalidate>
     <h1>Login</h1>
     <p>Log in to access the site</p>
     <input type="email" name="email" placeholder="Email"
autofocus
     ng-model="user.email" ng-required="true">
     <input type="password" name="password"
placeholder="password"
     ng-model="user.password" ng-required="true">
     <button type="submit" class="btn">Login</button>
     <p>or <a href="#/register">Register</a></p>
   </form>
</section>
```

✦ add ng-show - login.html

```
<section class="login">
   <form name="myform" novalidate>
     <h1>Login</h1>
     <p>Log in to access the site</p>
     <input type="email" name="email" placeholder="Email"
autofocus
     ng-model="user.email" ng-required="true">

     <p class="error"
     ng-show="myform.email.$invalid && myform.email.$touched">
     Must be a valid email
     </p>

     <input type="password" name="password"
placeholder="password"
     ng-model="user.password" ng-required="true">

     <p class="error"
     ng-show="myform.password.$invalid && myform.password.
$touched">
     Must enter a password
     </p>

     <button type="submit" class="btn">Login</button>
     <p>or <a href="#/register">Register</a></p>
   </form>
</section>
```

✦ examine the form element in the inspector to see angular’s classes - login.html

```
<button type="submit" class="btn"
ng-disabled="myform.$invalid">Login</button>
```

Create Controllers for Login and Register

✦ register them in the routes - app.js

```
graphApp.config(['$routeProvider', function($routeProvider) {
    $routeProvider.
    when('/graph',{
        templateUrl: 'partials/graph.html',
        controller: 'graphControllers'
    })
    .when('/login',{
        templateUrl: 'partials/login.html',
        controller: 'registration'
    })
    .when('/register',{
```

```
        templateUrl: 'partials/register.html',
        controller: 'registration'
    })
    .otherwise({
     redirectTo: '/login'
    });
}]);
```

✦ add registration.js and link to in index.html

```
<script src="app/controllers/controllers.js"></script>
<script src="app/controllers/registration.js"></script>
```

✦ then add - registration.js

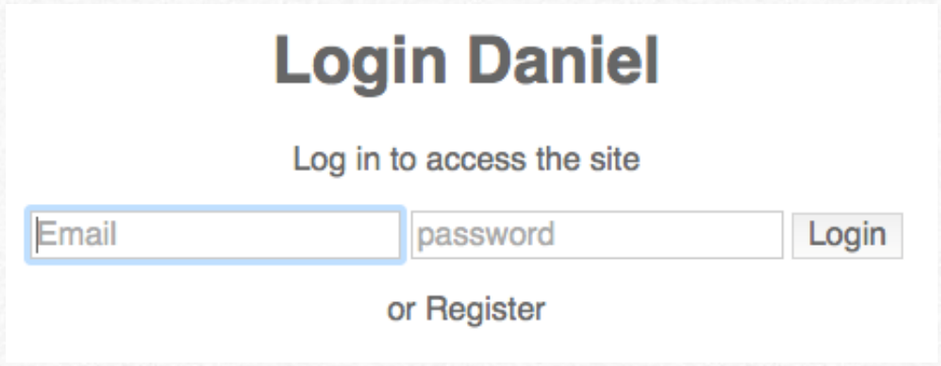
```
graphApp.controller('registration', function($scope){
});
```

✦ test it with a variable (controllers.js)

```
graphApp.controller('registration', function($scope){
  $scope.name="Daniel";
});
```

✦ in partial - login.html

```
<section class="login">
   <form name="myform" novalidate>
      <h3>Login {{name}}</h3>
      <p>Log in to access the site</p>
```

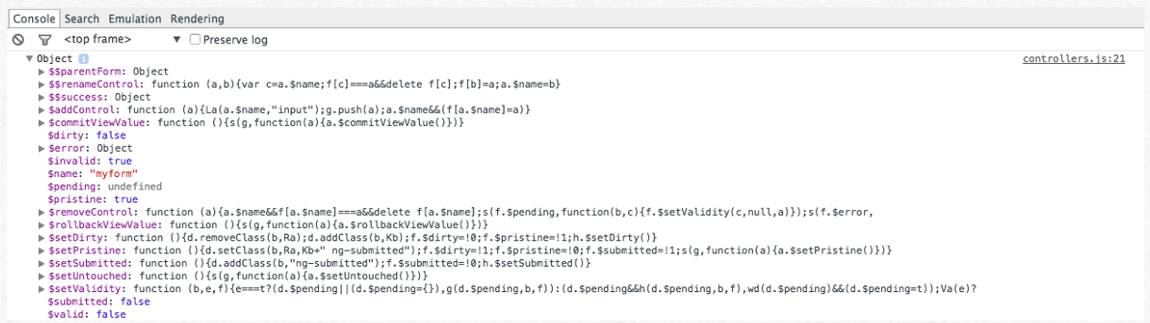


✦ try

```
graphApp.controller('registration', function($scope){
  $scope.name="Daniel";
  console.log($scope.myform);
});
```

✦ the console reads undefined because the page hasn't finished loading (async)

```
graphApp.controller('registration', function($scope){
   $scope.name="Daniel";
   $scope.$on("$viewContentLoaded", function(){
     console.log($scope.myform);
   });
});
```



✦ do something with the form - login.html

```
<section class="login">
   <form name="myform" novalidate ng-submit="login()">
      <h1>Login</h1>
```

✦ and in the controller - registration.js

```
graphApp.controller('registration', function($scope){
  $scope.login = function() {
    alert($scope.user.email);
  }
});
```

✦ test in browser

Automatically take the user to the graphs page when login is good - review the docs for $location

```
graphApp.controller('registration',
  function($scope, $location){
    $scope.login = function() {
      $location.path("/graph")
    }
  });
```

✦ test and check docs for services - https://docs.angularjs.org/api/ng/service/$location

CSS Time

✦ in the scss file (also add the micro clearfix hack)

```
@import url('normalize.css');
/* Import fonts */
@import url("http://fonts.googleapis.com/css?family=Bree+Serif|Merriweather:400,300,700,900|Droid+Sans:400,700");

/* apply a natural box layout model to all elements, but allowing components to change */
html {
  box-sizing: border-box;
  background-image: -webkit-linear-gradient(top, #023E54, #10AAC0);
  min-height: 100%;
  height: auto;
}
*, *:before, *:after {
  box-sizing: inherit;
}
body {
  font-family: Merriweather, serif;
  font-size: 1rem;
  line-height: 160%;
  color: #666;
  text-align:center;
}
```

✦ at the bottom

```
nav {
  width: 100%;
  background: #EFC94C;
}
form {
  width: 80%;
  background: #FCF4DC;
  min-width: 310px;
  max-width: 700px;
  padding: 10px 20px;
  margin: 20px auto 0 auto;
  text-align: center;
  border-radius: 10px;
  box-shadow: 5px 5px 20px rgba(0,0,0,0.2);
  position: relative;
  opacity: .95;

  input {
    display: block;
    width: 90%;
    margin: 20px;
    padding: 6px;
  }
  button {
    display: inline-block;
    background: #14566A;
    color: #fff;
    margin: 0 auto;
    margin-top: 10px;
    padding: 6px 12px;
    border-radius: 6px;
```

```
    text-align: center;
    font-size: 1em;
    border: 2px solid #fff;
  }
}
```

✦ edit chart styles

```
.chart {
  color: #fff;
  border-left:1px solid #fff;
  border-bottom: 1px solid #fff;
  margin: 40px auto 0 auto;
  position: relative;
```

Registration

start by copying the contents of login partial to register and make the following additions to the existing email and password inputs:

```
<section class="register">
  <form name="myform" novalidate ng-submit="register()">
    <h1>Register</h1>
    <p>To access the site</p>

<input type="firstname" name="firstname" placeholder="first
name" autofocus
ng-model="user.firstname" ng-required="true">

<p class="error"
ng-show="myform.firstname.$invalid && myform.firstname.
$touched">
Must enter a first name
</p>

<input type="lastname" name="lastname" placeholder="lastname"
ng-model="user.lastname" ng-required="true">
```

```
<p class="error"
ng-show="myform.lastname.$invalid && myform.lastname.
$touched">
Must enter a last name
</p>
...
```

✦ test link in browser

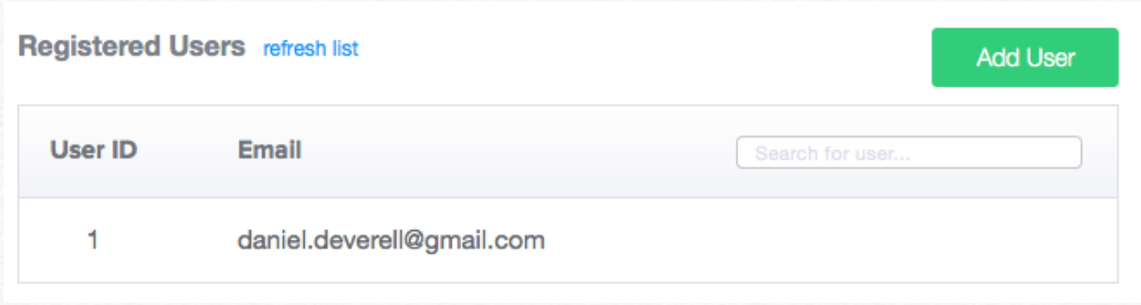
✦ set the location - registration.js

```
graphApp.controller('registration',
  function($scope, $location){
    $scope.login = function() {
      $location.path("/graph")
    }  //login

    $scope.register = function() {
      $location.path("/graph")
    }  //register
  });
```

Firebase registration service

✦ create a registered user



Adding a Constant

✦ in the app.js

```
var graphApp = angular.module("graphApp",["ngRoute",
"appControllers", "firebase"])
```

```
.constant('FIREBASE_URL', 'https://vizapp.firebaseio.com');
```

✦ in registration.js (NOTE: we also use the data in controllers.js)

```
graphApp.controller("registration", function($scope,
$location, FIREBASE_URL){
   var ref = new Firebase (FIREBASE_URL);
```

Additions in red below are taken directly from the user guide.

firebase auth - https://www.firebase.com/docs/web/libraries/angular/quickstart.html

firebase auth with password - https://www.firebase.com/docs/web/libraries/angular/api.html#angularfire-users-and-authentication-authwithpasswordcredentials-options

- registration.js

```
graphApp.controller("registration", function($scope,
$location, FIREBASE_URL, $firebaseAuth){
   var ref = new Firebase (FIREBASE_URL);
   var auth = $firebaseAuth(ref);

   $scope.login = function(){
      auth.$authWithPassword({
         email: $scope.user.email,
         password: $scope.user.password
      }).then(function(user){
         $location.path("/graph");
      }).catch(function(error){
         $scope.message = error.message;
      });
   } // login

   $scope.register = function() {
      //$location.path("/graph")
   }
});
```

✦ and add paragraph to the login.html file with error message handling - login.html

```
<h3>Login</h3>
<p>Login to enjoy the goodness.</p>
<p class="error" ng-show="message">{{ message }}</p>
```

✦ test the errors

✦ angular promises - http://andyshora.com/promises-angularjs-explained-as-cartoon.html

Custom Authentication Service

So that we can access our authenticated status not just from one page but from the entire site. Authentication is not specific to a single page. Every time you load up a page/view authentication is discarded.

We need to create a factory: http://tylermcginnis.com/angularjs-factory-vs-service-vs-provider/ We need to abstract our registration into a factory to make it available throughout the site.

✦ create a folder called services in app directory

✦ create new authentication.js file

✦ add it to the list in index

```
<script src="app/app.js"></script>
<script src="app/services/authentication.js"></script>
<script src="app/controllers/controllers.js"></script>
```

✦ edit authentication.js to create factory so that we can call it in other controllers - authentication.js

```
graphApp.factory('Authentication', function(
   $firebase, $firebaseAuth, $routeParams, $location,
FIREBASE_URL) {
```

```
}); // factory
```

✦ add two variables - authentication.js

```
graphApp.factory('Authentication', function(
   $firebase, $firebaseAuth, $routeParams $location,
FIREBASE_URL) {

   var ref = new Firebase (FIREBASE_URL);
   var auth = $firebaseAuth(ref);

}); // factory
```

✦ create an object that has some methods (review eloquent js?). When we call this function from the registration controller it passes along a variable called user. N.B. - authWithPassword is a method.

✦ - **authentication.js**

```
graphApp.factory('Authentication', function(
   $firebase, $firebaseAuth, $routeParams $location,
FIREBASE_URL) {

   var ref = new Firebase (FIREBASE_URL);
   var auth = $firebaseAuth(ref);

   var myObject = {
      login: function(user){
         return auth.$authWithPassword({
            email: user.email,
            password: user.password
         }); // authWithPassword
      } //login
   }; // myObject
   return myObject;
}); // factory
```

✦ edit registration js. The user is coming in from the view (examine login.html for the info) - registration js

```
graphApp.controller("registration",
   function($scope, $location, FIREBASE_URL, $firebaseAuth,
Authentication){

   var ref = new Firebase (FIREBASE_URL);
   var auth = $firebaseAuth(ref);

   $scope.login = function(){
      Authentication.login($scope.user)
      .then(function(user){
         $location.path("/graph");
      }).catch(function(error){
         $scope.message = error.message;
      });
   } // login

   $scope.register = function() {
      //$location.path("/graph")
   }
});
```

✦ test in browser - should see “The specified user does not exist.” for a new user. Should be forwarded to graphs if you use a real user.

Registering Users

✦ copy and paste the login function and make changes - **registration.js**

```
graphApp.controller("registration",
   function($scope, $location, FIREBASE_URL, $firebaseAuth,
Authentication){

   var ref = new Firebase (FIREBASE_URL);
   var auth = $firebaseAuth(ref);

   $scope.login = function(){
      Authentication.login($scope.user)
      .then(function(user){
```

```
      $location.path("/graph");
    }).catch(function(error){
      $scope.message = error.message;
    });
  }; // login
////////////////////////////////////
  $scope.register = function() {
// create a register method and pass the user info (an object)
in scope. examine registration.html
    Authentication.register($scope.user)
// after that we receive a promise with the user data in it
    .then(function(user) {
// now log them in and call the authentication service and set
the path
      Authentication.login($scope.user);
      $location.path("/graph");
// catch function - http://www.w3schools.com/js/js_errors.asp
    }).catch(function(error){
      $scope.message = error.message;
    });
  }; // register
});
```

✦ copy and paste the error message from login.html to - **register.html**

```
<section>
  <form name="myform" novalidate ng-submit="register()">
    <h3>Register v9</h3>
    <p>Register to enjoy the goodness.</p>
    <p class="error" ng-show="message">{{ message }}</p>
```

✦ create a register function in - **authentication.js**

```
graphApp.factory('Authentication', function(
  $firebase, $firebaseAuth, $routeParams $location,
FIREBASE_URL) {

  var ref = new Firebase (FIREBASE_URL);
  var auth = $firebaseAuth(ref);

  var myObject = {
    login: function(user){
      return auth.$authWithPassword({
        email: user.email,
        password: user.password
      }); // authWithPassword
    }, //login
  }; // myObject
  return myObject;
}); // factory
```

Since we are working with an ***object*** (objects are variables containing variables) we need to add a comma first. The object gets returned to whatever calls that object. The login is technically a method (methods are actions that can be performed on objects) - http://www.w3schools.com/js/js_object_methods.asp

✦ use the firebase method $createUser -- authentication.js

```
graphApp.factory('Authentication', function(
  $firebase, $firebaseAuth, $routeParams, $location,
FIREBASE_URL) {

  var ref = new Firebase (FIREBASE_URL);
  var auth = $firebaseAuth(ref);

  var myObject = {
    login: function(user) {
      return auth.$authWithPassword({
        email: user.email,
        password: user.password
      }); // authWithPassword
    }, //login

    register: function(user) {
```

```
        return auth.$createUser({
          email: user.email,
          password: user.password
        });
      } // register
    }; // myObject
    return myObject;
}); // factory
```

✦ **test adding a new user in browser.** Note that the **firebase** users page needs to be refreshed. Should forward to graphs.

✦ Check out the authentication in **firebase**.

Storing User Data in Firebase

✦ execute a function in the authentication service (in the register section) - **authentication.js**

```
register: function(user) {
  return auth.$createUser({
    email: user.email,
    password: user.password
  }).then(function(authData){

  });
} // register
```

✦ we need to add a different data area to our firebase project - **authentication.js.**

```
register: function(user) {
  return auth.$createUser({
    email: user.email,
    password: user.password
  }).then(function(regUser){
    var ref = new Firebase(FIREBASE_URL);
    var postRef = ref.child('users').child(authData.uid);
    postRef.set({
        date: Firebase.ServerValue.TIMESTAMP,
        firstname: user.firstname,
        lastname: user.lastname,
        email: user.email,
        password: user.password
      });
    });
  } // register
  } // myObject
```

✦ NB - check the FIREBASE_URL constant in **app.js** and make adjustments (**controllers.js**) and deletions (**registration.js**)

✦ see firebase reference for $set() -

✦ delete users and **test in browser** - test with duplicate user (already registered)

✦ be sure to examine both data and registered users tabs in firebase NB - you can have a user without data at this point.

```
Notes:
firebase had angular specific info here:
https://www.firebase.com/docs/web/libraries/angular/
api.html#angularfire-firebaseobject

and more general info (e.g. on set() here:
https://www.firebase.com/docs/web/guide/saving-data.html
```