Angular

by Jason Swett

Day 1 Overview

- Intros
    - About Me
    - About You
- Intro to Angular
    - Why Angular?
    - Lab: Angular Hello World
- TypeScript
    - ES5/ES6/TypeScript Differences
    - Decorators
    - Lab: ES5 to ES6 to TypeScript

Day 1 Overview

- Directives
    - Structural Directives
    - Attribute Directives
    - @Input
    - Lab: Write a Directive
- Modules
    - Imports
    - Declarations
    - Providers
    - Bootstrap
- Dependency Injection

Day 1 Overview

- Components
    - Lab: Write a Component
- Data Binding
    - Interpolation
    - Property Binding
    - Variables
    - Template Statements
    - Two-Way Binding
    - Lab: Data Binding

About Me

- Jason Swett
- Consultant, Trainer, Mentor
- Author of *Angular for Rails Developers*, Ruby Rogues panelist
- Coding since about 1996
- Background is Angular, Rails, and before that, PHP and other technologies
- I live in Sand Lake, Michigan, USA
- 33 years old, wife and two kids (4 and 6)

Where I Live


WASHINGTON
MONTANA
NORTH DAKOTA
MINNESOTA
SOUTH DAKOTA
WISCONSIN
MICHIGAN
Ottawa
Montreal
NB
PE
NOVA SCOTIA
MAINE
VT
NH
MA
CT RI
NEW YORK
Toronto
OREGON
IDAHO
WYOMING
Chicago
IOWA
NEBRASKA
ILLINOIS
INDIANA
OHIO
PENN
Philadelphia
MD
DE
NJ
WEST VIRGINIA
KENTUCKY
VIRGINIA
NORTH CAROLINA
SOUTH CAROLINA
TENNESSEE
ARKANSAS
MISSISSIPPI
ALABAMA
GEORGIA
FLORIDA
LOUISIANA
Houston
Dallas
OKLAHOMA
TEXAS
NEW MEXICO
ARIZONA
COLORADO
KANSAS
MISSOURI
UTAH
NEVADA
CALIFORNIA
San Francisco
Las Vegas
Los Angeles
San Diego
United States
Mexico
Gulf of California
Gulf of Mexico

Where I Live

My Family

About You

- Your name
- Brief technology background
- Angular experience (if any)
- What you're hoping to get out of this class

Class Material

- https://github.com/jasonswett/may-2017-angular-class

Why Angular?

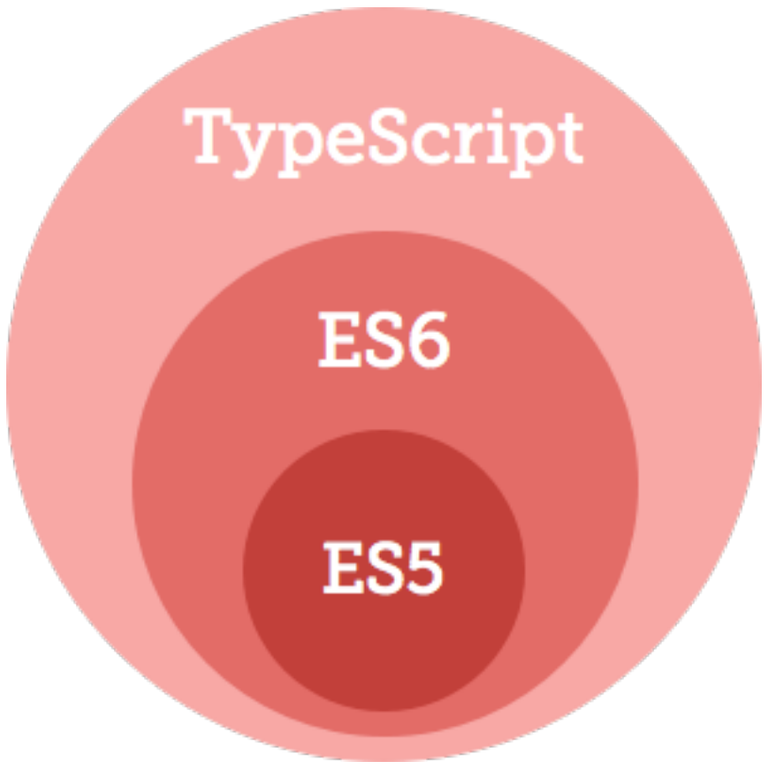
- Provides structure on the front-end (frameworks like jQuery provide some tools but not much structure)
- Richer UI functionality (example: Gmail)
- Faster (potentially)
- Decoupled front-end/back-end development

Lab: Angular Hello World

- Clone the Angular QuickStart repo at `https://github.com/angular/quickstart`

- Run `npm start` to run the server

TypeScript



TypeScript

- “TypeScript is a typed superset of JavaScript that compiles to plain JavaScript”
- Created by Microsoft
- TypeScript is a superset of ES6 which is a superset of ES5
- Not supported by browsers, so must be transpiled

ES5/ES6 Class Syntax

ES5:

```
function Book(name) {

  this.name = name;

}
```

ES6:

```
class Book {

  constructor(name) {

    this.name = name;

  }

}
```

ES6/TypeScript Class Syntax

ES6:

```
class Book {

  constructor(name) {

    this.name = name;

  }

}
```

TypeScript:

```
class Book {

  constructor(public name: string) {}

}
```

Decorators

- A TypeScript thing, not an Angular thing
- “A Decorator is a special kind of declaration that can be attached to a class declaration, method, accessor, property, or parameter. Decorators use the form @expression, where expression must evaluate to a function that will be called at runtime with information about the decorated declaration.”

Decorator Example

```
@Directive({
  selector: '[myHighlight]'
})
export class HighlightDirective {}
```

TypeScript Compiler (tsc)

- Transpiles TypeScript into ES5

- `npm install -g typescript`

ES6 Module Loading Syntax

```
export class HighlightDirective {}
```

```
import { HighlightDirective } from `./highlight.directive';
```

Doesn’t have good browser support yet

Lab: ES5 to ES6 to TypeScript

1. Create a tiny program in ES5

2. Modify the ES5 program to take advantage of ES6 features

3. Modify the ES6 program to take advantage of TypeScript features

4. See how `tsc` transpiles TypeScript to ES5

Lab: ES5 to ES6 to TypeScript

- Two concepts: an Author with a name and a list of books, and a Book with a title and a year of publication
- No UI, just send output to the log
- I'll go through it, then I'll have you go through it

Directives

- One of the main building blocks of Angular, especially when built as a component

- There are three types of directives

Directives

- Components—directives with a template.
- Structural directives—change the DOM layout by adding and removing DOM elements.
- Attribute directives—change the appearance or behavior of an element, component, or another directive.

Structural Directive Example

```
<ul>
  <li *ngFor="let book of books">{{ book.name }}</li>
</ul>
```

Attribute Directive Example

```
<p myHighlight>Highlight me</p>
```

@Directive Decorator

```
@Directive({
  selector: '[myHighlight]'
})
export class HighlightDirective {}
```

@Input

- A decorator that allows values to be passed from template to directive

@Input

In directive:

```
@Input() highlightColor: string;
```

In markup:

```
<p myHighlight
highlightColor="yellow">Highlighted in
yellow</p>
```

Lab: Write an Attribute Directive

Overview:

1. Clone the Angular QuickStart repo at `https://github.com/angular/quickstart` and run `npm install`
2. Get a directive called `myHighlight` to show up on the page
3. Change the directive to respond to user-initiated events
4. Pass values into the directive with `@Input`
5. Bind to a second property

Lab: Write an Attribute Directive (1)

- Go to https://angular.io/docs/ts/latest/guide/attribute-directives.html (google “angular 2 attribute directives”)

- Complete the first step

Lab: Write an Attribute Directive (2)

- Complete the steps under “Respond to user-initiated events”

- Text should be highlighted as you hover over it

Lab: Write an Attribute Directive (3)

- Complete the steps under “Pass values…”

- Radio buttons should be present and text should highlight according to which radio button is selected

Lab: Write an Attribute Directive (4)

- Complete the steps under “Bind to a second property”
- The two pieces of text should have different default colors

Modules

- Allow you to make your Angular application more modular

Modules

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [
    AppComponent,
    HighlightDirective
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

Module is Defined as a Class

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [
    AppComponent,
    HighlightDirective
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

@NgModule Decorator Contains Config for the Module

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [
    AppComponent,
    HighlightDirective
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

imports

Makes the exported declarations of other modules available in the current module

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [
    AppComponent,
    HighlightDirective
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

(BrowserModule provides services and directives like ngIf, ngFor)

declarations

Makes directives in the current module available to other directives in the current module

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [
    AppComponent,
    HighlightDirective
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

providers

Makes services and other values **injectable**

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [
    AppComponent,
    HighlightDirective
  ],
  providers: [ MyService ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

Dependency Injection

Dependency Injection

- Global entities mean implicit dependencies
- Global entities can be get or set by any part of the system
- Global entities means namespace pollution and potential naming conflicts
- Dependency injection addresses these issues by requiring all dependencies to be explicitly declared

Dependency Injection

```
import { Component } from '@angular/core';
import { MyService } from './my.service';

@Component({
  selector: 'my-app',
  templateUrl: './app.component.html'
})
export class AppComponent {
  constructor(private myService: MyService) {}
}
```

Components

Components

- Component = Directive + Template

- One of the main building blocks of Angular

Lab: Write a Component

1. Clone the Angular QuickStart repo at `https://github.com/angular/quickstart` and run `npm install`

2. Create a `MyButtonComponent` in a new file called `my-button.component.ts`

3. Get your button to show up on the page

Data Binding

Data Binding: Interpolation

- Interpolation ({ {...} })
- `<p>My current hero is {{currentHero.name}}</p>`
- https://angular.io/docs/ts/latest/guide/template-syntax.html

Data Binding: Property Binding

- `[property]="expression"`

- `<span [hidden]="isUnchanged">changed</span>`

Data Binding: Variables

- Template input variable:
  - `<div *ngFor="let hero of heroes”>{{hero.name}}</div>`
- Template reference variable:
  - `<input #heroInput> {{heroInput.value}}`

Data Binding: Template Statements

- `<button (click)="onSave($event)">Save</button>`
- `<button *ngFor="let hero of heroes" (click)="deleteHero(hero)">{{hero.name}}</button>`
- `<form #heroForm (ngSubmit)="onSubmit(heroForm)"> ... </form>`

Data Binding: Two-Way Binding

```
<input [value]="name" (input)="name = $event.target.value">
```

- `[value]="name"` - Binds the expression `name` to to input element's value property
- `(input)=""` - Binds an expression to the input element's `input` event
- `$event` - An expression exposed in event bindings by Angular, which has the value of the event's payload
- `name = $event.target.value` - The expression that gets evaluated when the `input` event is fired
- The above comes from https://blog.thoughtram.io/angular/2016/10/13/two-way-data-binding-in-angular-2.html

Data Binding

```
<input
[ngModel]="name" (ngModelChange)="name =
$event”>
```

- `ngModel` requires `FormsModule`
- `[ngModel]=“name”` is similar to `[value]=“name”`
- `(ngModelChange)` is similar to `(input)`

Data Binding

```
<input [(ngModel)]="name">
```

- “Banana in a box” shorthand

Lab: Data Binding

1. “Manually” bind a component property to an input
2. Bind a property to an input using ngModel
3. Bind a property to an input using “banana-in-a-box” ngModel shorthand

Break

Lab: Contact Manager

- Two inputs: Name and Phone
- Each contact entered should be added to a list on the page
- Bonus: get contacts to save to localStorage
- Bonus 2: refactor application to use a service

Day 1 Review

- Intro to Angular
  - Why Angular?
  - Lab: Angular Hello World
- TypeScript
  - ES5/ES6/TypeScript Differences
  - Decorators
  - Lab: ES5 to ES6 to TypeScript

Day 1 Review

- Directives
    - Structural Directives
    - Attribute Directives
    - @Input
    - Lab: Write a Directive
- Modules
    - Imports
    - Declarations
    - Providers
    - Bootstrap
- Dependency Injection

Day 1 Review

- Components
    - Lab: Write a Component
- Data Binding
    - Interpolation
    - Property Binding
    - Variables
    - Template Statements
    - Two-Way Binding
    - Lab: Data Binding

Evaluation Time

https://goo.gl/gEZ9sd