

ANGULARJS COURSE CONTENT

- Introduction to AngularJS
 - What does AngularJS do for me?
 - Who controls AngularJS?
 - How can I get AngularJS?
- Our first AngularJS application
 - A basic application
 - Using angular-seed
 - The pieces of the puzzle
 - Two-way data binding
 - Directives
 - How it fits together
 - How much of the page is an Angular application?
 - Model, View, Controller from the AngularJS Perspective
- Controllers
 - Where Controllers fit in, and what they do, from Angular's perspective
 - Managing Scope
 - Setting up Behavior
 - Building a basic controller
 - A more advanced controller
- Models
 - How to create a model
 - Explicit models
 - Implicit models
- Views
 - Angular's take on the View: a little bit different
 - Tying a View to a Controller
 - Tying a View to a model

- Expressions
 - Expressions are lightweight code snippets
 - Expression capabilities
 - Limitations
 - The border between expressions and `$eval`
- Filters
 - Standard filters
 - Writing your own filter
 - Tying filters together
- Scopes
 - What are scopes?
 - What do scopes provide?
 - Scope lifecycle
 - Scopes as glue between controller and view
 - Scope hierarchies
 - Scope and events
- Angular Forms
 - Angular forms vs HTML forms
 - Angular form controls
 - Form events
 - The form controller
 - Form validation
 - CSS classes for form data
- Ajax, Data, and Angular
 - High level interactions with servers
 - Low-level server interactions with `$http`
 - The deferred/promises API
- Directives
 - Teaching HTML new tricks
 - Binding text and attributes
 - Directive processing lifecycle
 - DOM Processing
 - Compilation

- Linking
- A basic directive
- Directives and scopes
- Creating reusable directives
- Turning directives into components
- Testing in Angular
 - Unit testing
 - Working with Dependency Injection
 - Other unit testing issues
 - End-to-end testing
 - Angular's E2E testing framework
 - Commands and expectations
 - Controlling what happens before and after the test
 - Running a scenario