Angular

Operators

{{ }} - bindings operator

Modules

Why modules?

To make code maintainable, testable, readable.

What is a module?

A collection of services, directives, controllers, filters, and configuration info

What do modules have to do with dependencies?

Modules should be written such that dependencies are well defined and encapsulated in separate modules

Each module should encapsulate specific behavior, pulling in other modules where needed.

How do you create a module?

var app = angular.module('store', []);

[] - dependencies

'store' - name

app - module created with the angular.Module API

API

angular.module - Used to create and register a module

module.controller - Used to register a controller

Controllers

Defines application behavior by defining functions and values

Help to get data onto the page

Provides behavior

Directives

Anything in the quotes is already considered an expression, so no need to use brackets

ng-app - attached to html tag to create an angular application

references an angular application module

the module gets executed when the document is loaded

Causes html within the html tag to be considered as part of an angular app

ng-controller

Used to attach a controller to an html tag

<div ng-controller="LongNamedController as ctrl">

LongNamedController - name of controller defined in module

as - keyword used when defining aliases

ctrl - aliases

ng-show

Used to only show content if an expression is true

ng-hide

Similar to ng-show

ng-repeat

iterate over

ng-src

causes angular to parse an expression into a url to be retrieved by the browser

<img ng-src="http://mg.com/{{store.products[0].image}}">

ng-click

assigns a value to a variable on clicking an element

any expression in the page which uses that variable will get updated

ng-init

allows an expression to be evaluated in the current scope.

use to initialize data

really meant for poc's

usually initialization logic should go in a controller

ng-class

adds a class if a given expression is true

ng-class="{ active:tab === 3 }"

ng-model

binds form elements to controller properties (model)

ng-submit

causes a function to be invoked when a form is submitted

Services

Provide a controller with additional functionality

Basically like a jQuery plugin.

Prepackaged bunches of functionality

How do you use serivces?

Need to let your controller know what services it needs

Get passed as args to your controller, giving your controller access to them

This is considered dependecy injection

Expression

Javascript-like expressions

How are angular expressions different from javascript expressions?

context

javascript - evaluated w respect to global window object

angular - evaluated w respsect to a scope object

prevents direct access to window, document, location

forgiving

js - ReferenceError / TypeError are thrown when trying to evaluate undefined props

angular - Errors aren't thrown when trying to evaluate undefined props

no control flow statements

angular - loops, conditionals and exceptions aren't allowed

ternary is allowed

Filters

angular - output can be filtered

How can complex expressions be evaluated in angular?

Move the expression to a controller method

What's the difference between { } and {{ }}?

They are completely different.

double brackets are expressions.

single brackets are used for js objects

Numeric operations

{{4 + 6}}

String operations

{{"hello" + " you"}}

http://docs.angularjs.org/guide/expression

Filter

{{ data | filter:options}}

currency

{{ price | currency }}

date

{{ '2343242342423' | date:'MM/dd/yyyy @ h:mma' }}

uppercase / lowercase

{{ 'foo' | uppercase }}

limitTo

{{ 'Testing' | limitTo:4 }}

<li ng-repeat="product in store.products | limitTo:1">

orderBy

<li ng-repeat="product in store.products |orderBy:'-price'">

2-way data-binding

expressions get reevaluated whenever a property changes

Terms

Angular - A js library written by google

Module - Used to encapsulate code.

Directive - elements within an html page which bind behavior to structure

Expression - Used to insert dynamic values into DOM

Data - Variable, js object

Filter - TODO

localized - values are formatted according to the current region.

Closure -

An anonymous function not bound to an identifier.

Defines a scope within which variables may be defined and used but not externally accessed, save by functions defined within and returned from the closure

Defines a scope within which variables present at the time the closure is defined are accessible even after the scope the closure was defined in has been destroyed.

A stack-frame which isn't destroyed when the caller's scope is destroyed

Variables since made out of scope are still available in the closure

free variable - A variable only accessible within the scope of a closure or by references established within and / or returned from a closure

Anonymous function - Another name for a closure

Inner function - Todo

An anonymous function

Injector - Essentially a registry used for dependency injection

stack-frame - Todo

Promise - like a future, but in jQuery