

### Directives

**ng-app**="plaintext"

**ng-bind**[**html-unsafe**]="expression"

**ng-bind-template**="string"

**ng-change**="expression"

**ng-checked**="boolean"

**ng-class**[**-event-odd**]="string|object"

**ng-[dbl]click**="expression"

**ng-cloak**="boolean"

**ng-controller**="plaintext"

<html **ng-csp**> (Content Security Policy)

**ng-disabled**="boolean"

<form**ng-form** name="plaintext"> | **ng-form**="plaintext"

**ng-hide**|**show**="boolean"

**ng-href**="plaintext{{{string}}}"

**ng-include**="string"|<**ng-include** src="string" *onload*="expression" *autoscrol*="expression">

**ng-init**="expression"

<input **ng-pattern**="/regex/" **ng-minlength** **ng-maxlength** **ng-required**

<input **ng-list**="delimiter|regex">

**ng-model**="expression"

**ng-mouse**[**down**|**enter**|**leave**|**move**|**over**|**up**]="expression"

<select **ng-multiple**>

**ng-non-bindable**

**ng-options**="select [*as label*] [*group by group*] for ([*key*,] value) in object[array]"

**ng-pluralize**|<**ng-pluralize** count="number" when="object" *offset*="number">

**ng-readonly**="expression"

**ng-repeat**="([*key*,] value) in object[array]"

<option **ng-selected**="boolean">

**ng-src**="string"

**ng-style**="string|object"

**ng-submit**="expression"

**ng-switch**="expression"|<**ng-view** on="expression">

**ng-transclude** templates

**ng-view**|<**ng-view**>

**ng-bind-html**="expression"

**Bold** means the actual directive

*Italics* mean optional

Pipes mean either|or

Plaintext means no string encapsulation

Superscript means notes or context

<Brackets> mean tag compitibility

Lack of <brackets> means the attribute can apply to any tag

### Filters

**amount** | **currency**[:*symbol*]

| Formats a number as a currency (ie \$1,234.56).

**date** | **date**[:*format*]

**array** | **filter**:expression

| Selects a subset of items from array. Expression takes *string*|*Object*|*function*()

**data** | **json**

| Convert a JavaScript object into JSON string.

**array** | **limitTo**:limit

| Creates a new array containing only a specified number of elements in an array.

**text** | **linky**

| Finds links in text input and turns them into html links.

**string** | **lowercase**

| Converts string to lowercase.

**number** | **number**[:*fractionSize*]

| Formats a number as text. If the input is not a number an empty string is returned.

**array** | **orderBy**:predicate[:*reverse*]

| Predicate is function(\*)|string|Array. Reverse is boolean

**string** | **uppercase**

| Converts string to uppercase.

You can inject the \$filter service and do *\$filter('filterName')(value[, .optionalParam][, .optionalParam])* in use it in your javascript.

### Services

**\$anchorScroll**

**\$cacheFactory**

compiledHtml = **\$compile**(html)(scope)

**\$controller**

**\$cookieStore**

**\$document**

**\$exceptionHandler**(exception[, *cause*])

**\$filter**(name)

**\$http**[[*options*]]

**\$httpBackend**

**\$injector**

**\$interpolate**(text[, *mustHaveExpression*])

**\$locale**

**\$location**

**\$log**

**\$parse**(expression)

**\$provide**

**\$q**

**\$resource**(url[, paramDefaults][, actions])

**\$rootElement**

**\$rootScope**

**\$route**

**\$routeParams**

**\$routeProvider**

**\$sanitize**(html)

### Global Functions

**angular.bind**(self, fn, args)

| Returns a function which calls function fn bound to self (self becomes the this for fn).

**angular.bootstrap**(element[, *modules*])

| Use this function to manually start up angular application.

**angular.copy**(source[, *destination*])

| Creates a deep copy of source, which should be an object or an array.

**angular.element**(element)

| Wraps a raw DOM element or HTML string as a jQuery element.

**angular.equals**(o1, o2)

| Determines if two objects or two values are equivalent.

**angular.extend**(dst, src)

| Extends the destination object dst by copying all of the properties from the src object(s) to dst.

**angular.forEach**(obj, iterator[, *context*])

| Invokes the iterator function once for each item in obj collection, which can be either an object or an array.

**angular.fromJson**(json)

| Deserializes a JSON string.

**angular.identity**()

| A function that returns its first argument. This function is useful when writing code in the functional style.

**angular.injector**(modules)

| Creates an injector function that can be used for retrieving services as well as for dependency injection.

**angular.isArray**(value)

| Determines if a reference is an Array.

**angular.isDate**(value)

| Determines if a value is a date.

**angular.isDefined**(value)

| Determines if a reference is defined.

**angular.isElement**(value)

| Determines if a reference is a DOM element (or wrapped jQuery element).

**angular.isFunction**(value)

| Determines if a reference is a Function.

**angular.isNumber**(value)

| Determines if a reference is a Number.

**angular.isObject**(value)

| Determines if a reference is an Object. Unlike typeof in JavaScript, nulls are not considered to be objects.

**angular.isString**(value)

| Determines if a reference is a String.

**angular.isUndefined**(value)

| Determines if a reference is undefined.

**angular.lowercase**(string)

| Converts the specified string to lowercase.

**angular.mock**

| Namespace from 'angular-mocks.js' which contains testing related code.

**angular.module**(name[, *requires*], configFn)

### Module

**config**(configFn)

| Use this method to register work which needs to be performed on module loading.

**constant**(name, object)

	Because the constant are fixed, they get applied before other provide methods.
<b>controller(name, constructor)</b>	
<b>directive(name, directiveFactory)</b>	
<b>factory(name, providerFunction)</b>	
<b>filter(name, filterFactory)</b>	
<b>provider(name, providerType)</b>	
<b>run(initializationFn)</b>	
	Use this method to register work which needs to be performed when the injector with with the current module is finished loading.
<b>service(name, constructor)</b>	
	value(name, object)
<b>name</b>	
	Name of the module.
<b>requires</b>	
	Holds the list of modules which the injector will load before the current module is loaded.


NgModelController	
\$render()	Called when the view needs to be updated. It is expected that the user of the ng-model directive will implement this method.
\$setValidity(validationErrorKey, isValid)	
\$setViewValue(value)	
\$viewValue	mixed
\$modelValue	mixed
\$parsers	array of function after reading val from DOM to sanitize / convert / validate the value
\$formatters	array of functions to convert / validate the value
\$error	object
\$pristine	boolean
\$dirty	boolean
\$valid	boolean
\$invalid	boolean

<b>\$scope</b>	See <i>\$rootScope</i>
<b>\$templateCache</b>	
<b>\$timeout</b> (fn[, delay][, invokeApply])	
<b>\$window</b>	

Scope Properties and Methods	
<b>\$root</b> or <b>\$rootScope</b>	
	Move to the top-most \$scope (ng-app)
<b>\$parent</b>	
	Move to the immediate parent of the current \$scope
<b>\$id</b>	
	Auto generated Unique ID
<b>\$destroy</b> (event)	
	Broadcasted when a scope and its children are being destroyed
<b>\$apply</b> (exp)	
	Executes logic within the AngularJS context and refreshes all models checks.
<b>\$broadcast</b> (name, args)	
	Dispatches an event name downwards to all child scopes
<b>\$destroy</b> ()	
	Removes the current scope (and all of its children) from the parent scope
<b>\$digest</b> ()	
	Process all of the watchers of the current scope and its children. Since watchers can change models, they will continue firing until all changes stop. <b>BEWARE OF RECURSIVE CODE</b>
<b>\$emit</b> (name, args)	
	Dispatches an event name upwards through the scope hierarchy
<b>\$eval</b> (expression)	
	Executes the expression on the current scope and returns the result
<b>\$evalAsync</b> (expression)	
	Executes the expression on the current scope at a later point in time
<b>\$new</b> (isolate)	
	Creates a new child scope
<b>\$on</b> (name, listener)	
	Listens on events of a given type
<b>\$watch</b> (watchExp, listener(newVal, oldVal, scope), objectEquality)	
	Watch a model (exp) for changes and fires the listener callback. Pass <i>true</i> as a third argument to watch an object's properties too.
<p>The following directives create child scopes: <i>ngInclude</i>, <i>ngSwitch</i>, <i>ngRepeat</i>, <i>ngController</i>, <i>uiIf</i>. Calls to the same <i>ngController</i> will create multiple instances and <b>do not</b> share scopes. Remember to traverse up the tree to affect <i>primitives</i> on the intended scope: <i>ng-click="\$parent.showPage=true"</i></p>	

	The angular.module is a global place for creating and registering Angular modules. Requires argument always creates a new module.
<b>angular.noop()</b>	
	A function that performs no operations.
<b>angular.toJson(obj[, pretty])</b>	
	Serializes input into a JSON-formatted string.
<b>angular.uppercase(string)</b>	
	Converts the specified string to uppercase.
<b>angular.version</b>	
	An object that contains information about the current AngularJS version.

FormController
\$pristine
\$dirty
\$valid
\$invalid
\$error

Cheatographer	
	<b>ProLoser</b> <a href="http://cheatography.com/proloser/">cheatography.com/proloser/</a> <a href="http://www.DeanSofer.com">www.DeanSofer.com</a>

Cheat Sheet
<p>This cheat sheet was published on 9th August, 2012 and was last updated on 28th October, 2012.</p>

Sponsor
<p><b>FeedbackFair</b>, increase your conversion rate today! Try it free!  <a href="http://www.FeedbackFair.com">http://www.FeedbackFair.com</a></p>