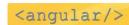
Hello World!

ng:change

ng:class



Hello World Example Code: <!doctype html> <html xmlns:ng="http://angularjs.org"> <script src="http://code.angularjs.org/angular-</pre> 0.9.19.min.js" ng:autobind></script> <body> Your name: <input type="text" name="yourname" value="World"/> <hr/>/ Hello {{yourname}}! </body></html> Result: The Hello World of angular X ← → C ☆ file:///path/to/hello_angular.html ▶ Your name: World

Widgets	
ng:non-bindable	Blocks angular from
	processing an HTML element.
ng:repeat	Creates a collection of cloned
	HTML elements
ng:required	Verifies presence of user
	input.
ng:validate	Validates content of user
	input
ng:view	Works with \$route to
	"include" partial templates.
ng:switch	Conditionally changes DOM
	structure.
ng:include	Includes an external HTML
	fragment.
ng:options	Dynamically generates a list
	of <i><option></option></i> elements for a
	<select> element using an</select>
	array or an object.
Directives	
ng:submit	Binds angular expressions to

onsubmit events.

on an element.

Runs an expression when an

input widget's value changes.

Conditionally set CSS class

ng:click	Executes custom behavior
	when element is clicked.
ng:controller	Creates a controller and links
	it to the DOM.
EU.	
Filters	
{{expression <	filter-name>:
<	param1>: <param2>:}}</param2>
currency	Formats a number as a
	currency.
date	Formats a number as a date.
	E.g. {{1288323623006
	date:'yyyy-MM-dd HH:mm:ss
	Z'}}
html	Prevents the input from getting
	escaped by angular.
json	Converts a JavaScript object
	into JSON string.
linky	Turns links into html links.
lowercase	Lowercases a string.
number	Formats a number as text
	Usage:{{number_expression
	number[:fractionSize] }}

Custom Filter angular.filter(<name>. function(input[,additional params]) { return uiValue;

ng:format=" <formatter-name>"</formatter-name>	
boolean	Formats user input as
	Boolean.
json	Formats user input as JSON.
list	Formats user input strings as
	an array.
number	Formats user input string as
	number.
trim	Trims leading and trailing
	spaces from user input.

Custom Formatter

```
angular.formatter(<name>, {
 parse: function(value) { ... },
 format: function(value { ... }
```

Validators

Formatters

ng:validate=" <validator< th=""><th>-nam</th><th>e>[:pa</th><th>ram1][:param</th><th>2]</th></validator<>	-nam	e>[:pa	ram1][:param	2]

[:]"	
asynchronous	Provides asynchronous
	validation via a callback
	function.

date	Checks if user input is a
	valid date.
email	Checks if user input is a
	valid email.
integer	Checks if user input is a
	valid integer.
Json	Checks if user input is a
	valid JSON.
number	Checks if user input is a
	valid number.
phone	Checks if user input is a
	valid phone number.
regexp	Restricts valid input to a
	specified regular
	expression pattern.
url	Validates that user input is
	a well formed URL.

Custom Validator

```
angular.validator(<name>, function(input
  [, additional params]) {
     [vour validation code]:
     if ([validation succeeds]) {
          return false;
     } else {
          return "my error message";
```

Services

\$defer(Defers function
callback[,delay])	execution.
\$invalidWidgets	Holds references to
	invalid widgets.
\$updateView	Queues view updates.
\$xhr(
\$xnr(Generates an XHR
method, url[, post],	Generates an XHR request.

Custom Service

```
angular.service(<name>.
  function(dep1) {
     return someService:
  }, {$inject: ['dep1']});
```

Controllers

<div <="" ng:controller="SomeController" th=""></div>
function SomeController(dep1) {}
SomeController.\$inject = ['dep1'];

this.\$watch(Registers listener as a
<watchexpr>,</watchexpr>	callback to be executed
function(value) {}	every time a watchExp
)	changes.

iQuery Mohile Angular Adapter

Juliery Mobile Angular Adapter		
ngm:shared- controller= "var1:Ctrl1,"	Directive to share the same instance of a controller between mobile pages. The shared controller is saved into a variable. E.g. <div ngm:shared-controller="rental:RentalController"></div>	
ngm: <event>= "action()"</event>	Directives for general event handlers. E.g. <a <br="" href="##">ngm:click="myFn()"> <a href="#" ngm:<br="">pageforeshow="myFn()">	
ngm:if	Conditionally changes DOM structure. Similar to ng:switch, but does not need nested elements.	

A service to access the

A service to change the

current page, optionally using a defined transition.

history to the defined page. pageId="back": Goes back one step in

transition="back": The browser will go back in

wait');

¡Query Mobile Wait Dialog.

this.\$waitDialog.show('Please

Navigation Expressions

\$navigate(test, 'outcome1[:transition1]:page1',...)

Specifies navigation based on outcomes. This will execute the navigation whose outcome equals test. If test is a promise, this will use the result of the promise.

history.

outcome="failure"

\$waitDialog:

- show([msg])

promise[, msg])

'[transition:]

- hide()

- waitFor(

\$navigate(

pageld')

Special outcome for the false value or a failed promise.

outcome="success"

Special outcome for all cases where "failure" does not match.

This can be used directly in HTML pages, e.g.

Paging for Lists

```
Lists can be paged in the sense that only some
entries of an array in the controller are shown.
ng:repeat="item in list.$paged()">
 {{item}}
ngm:click="list.$paged().loadNextPage()">
     Load more
 </a>
```



Kirchstraße 6 | 51647 Gummersbach Telefon 02261 6001-0 | Fax -4200

info@opitz-consulting.com www.opitz-consulting.com/go_mobile



www.youtube.com/opitzconsulting



www.slideshare.net/opitzconsulting



www.xing.com/net/opitzconsulting



www.twitter.com/oc_wire