AngularJS Expressions

AngularJS expressions can be written inside double braces: {{ *expression* }}.

AngularJS expressions can also be written inside a directive: ng-bind="*expression*".

AngularJS will resolve the expression, and return the result exactly where the expression is written.

**AngularJS expressions** are much like **JavaScript expressions:** They can contain literals, operators, and variables.

Example {{ 5 + 5 }} or {{ firstName + " " + lastName }}

Example

<!DOCTYPE html>  
<html>  
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>  
<body>  
  
<div ng-app="">  
  <p>My first expression: {{ 5 + 5 }}</p>  
</div>  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_expression)

If you remove the ng-app directive, HTML will display the expression as it is, without solving it:

Example

<!DOCTYPE html>  
<html>  
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>  
<body>  
  
<div>  
  <p>My first expression: {{ 5 + 5 }}</p>  
</div>  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_expression_2)

You can write expressions wherever you like, AngularJS will simply resolve the expression and return the result.

Example: Let AngularJS change the value of CSS properties.

Change the color of the input box below, by changing it's value:



Example

<div ng-app="" ng-init="myCol='lightblue'">  
  
<input style="background-color:{{myCol}}" ng-model="myCol" value="{{myCol}}">  
  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_expression_3)

AngularJS Numbers

AngularJS numbers are like JavaScript numbers:

Example

<div ng-app="" ng-init="quantity=1;cost=5">  
  
<p>Total in dollar: {{ quantity \* cost }}</p>  
  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_expressions)

Same example using ng-bind:

Example

<div ng-app="" ng-init="quantity=1;cost=5">  
  
<p>Total in dollar: <span ng-bind="quantity \* cost"></span></p>  
  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_bind_numbers)

|  |  |
| --- | --- |
| **Note** | Using ng-init is not very common. You will learn a better way to initialize data in the chapter about controllers. |

AngularJS Strings

AngularJS strings are like JavaScript strings:

Example

<div ng-app="" ng-init="firstName='John';lastName='Doe'">  
  
<p>The name is {{ firstName + " " + lastName }}</p>  
  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_expressions_strings)

Same example using ng-bind:

Example

<div ng-app="" ng-init="firstName='John';lastName='Doe'">  
  
<p>The name is <span ng-bind="firstName + ' ' + lastName"></span></p>  
  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_bind_strings)

AngularJS Objects

AngularJS objects are like JavaScript objects:

Example

<div ng-app="" ng-init="person={firstName:'John',lastName:'Doe'}">  
  
<p>The name is {{ person.lastName }}</p>  
  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_expressions_objects)

Same example using ng-bind:

Example

<div ng-app="" ng-init="person={firstName:'John',lastName:'Doe'}">  
  
<p>The name is <span ng-bind="person.lastName"></span></p>  
  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_bind_objects)

AngularJS Arrays

AngularJS arrays are like JavaScript arrays:

Example

<div ng-app="" ng-init="points=[1,15,19,2,40]">  
  
<p>The third result is {{ points[2] }}</p>  
  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_expressions_arrays)

Same example using ng-bind:

Example

<div ng-app="" ng-init="points=[1,15,19,2,40]">  
  
<p>The third result is <span ng-bind="points[2]"></span></p>  
  
</div>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_bind_arrays)

AngularJS Expressions vs. JavaScript Expressions

Like JavaScript expressions, AngularJS expressions can contain literals, operators, and variables.

Unlike JavaScript expressions, AngularJS expressions can be written inside HTML.

AngularJS expressions do not support conditionals, loops, and exceptions, while JavaScript expressions do.

AngularJS expressions support filters, while JavaScript expressions do not.