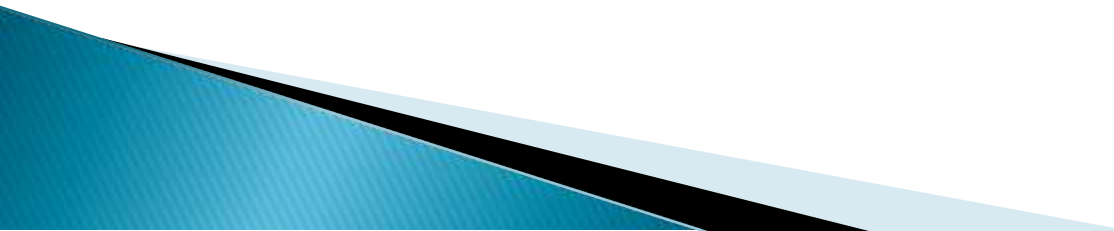


# ANGULAR JS

## MODULES

- ▶ An AngularJS module defines an application.
  - ▶ The module is a container for the different parts of an application.
  - ▶ The module is a container for the application controllers.
  - ▶ Controllers always belong to a module.
  - ▶ A module is created by using the AngularJS function
    - `angular.module`
- 

```
<div ng-app="myApp">...</div>
```

```
<script>
```

```
var app = angular.module("myApp", []);
```

```
</script>
```

- The "myApp" parameter refers to an HTML element in which the application will run.
- Now you can add controllers, directives, filters, and more, to your AngularJS application.

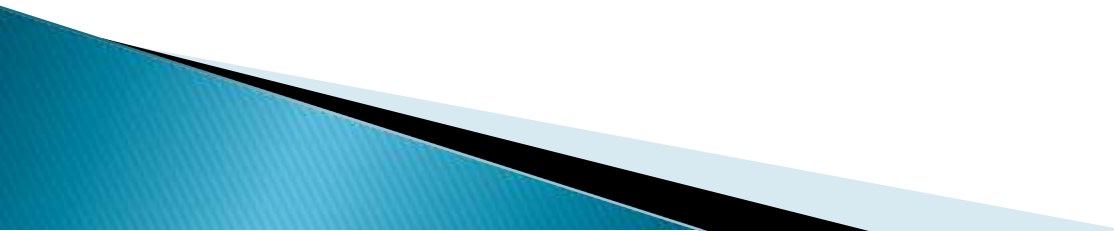
# Adding a Controller

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>

<div ng-app="myApp" ng-controller="myCtrl">
{{ firstName + " " + lastName }}
</div>

<script>
var app = angular.module("myApp", []);
app.controller("myCtrl", function($scope) {
    $scope.firstName = "John";
    $scope.lastName = "Doe";
});
</script></body></html>
```

# AngularJS Directives

- ▶ AngularJS directives are extended HTML attributes with the prefix ng-.
  - ▶ The ng-app directive initializes an AngularJS application.
  - ▶ The ng-init directive initializes application data.
  - ▶ The ng-model directive binds the value of HTML controls (input, select, textarea) to application data.
- 

```
<div ng-app="" ng-  
init="firstName='John'">
```

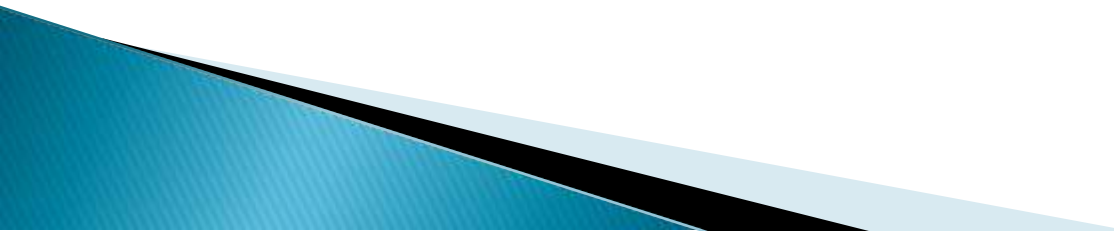
```
<p>Name: <input type="text" ng-  
model="firstName"></p>
```

```
<p>You wrote: {{ firstName }}</p>
```

```
</div>
```

- ▶ The `ng-app` directive also tells AngularJS that the `<div>` element is the "owner" of the AngularJS application.

# Data Binding

- ▶ The `{{ firstName }}` expression, in the example above, is an AngularJS data binding expression.
  - ▶ Data binding in AngularJS binds AngularJS expressions with AngularJS data.
  - ▶ `{{ firstName }}` is bound with `ng-model="firstName"`.
- 

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>

<div data-ng-app="" data-ng-init="quantity=1;price=5">

<h2>Cost Calculator</h2>

Quantity: <input type="number" ng-model="quantity">
Price: <input type="number" ng-model="price">

<p><b>Total in dollar:</b> {{quantity * price}}</p>

</div>

</body>
</html>
```



# Repeating HTML Elements

- ▶ The `ng-repeat` directive actually **clones HTML elements** once for each item in a collection.
- ▶ The `ng-repeat` directive used on an array of objects

```
<div ng-app="" ng-init="names=['Jani','Hege','Kai']">  
  <ul>  
    <li ng-repeat="x in names">  
      {{ x }}  
    </li>  
  </ul>  
</div>
```

```
<div ng-app="" ng-init="names=[
{name:'Jani',country:'Norway'},
{name:'Hege',country:'Sweden'},
{name:'Kai',country:'Denmark'}]">
```

```
<ul>
```

```
<li ng-repeat="x in names">
```

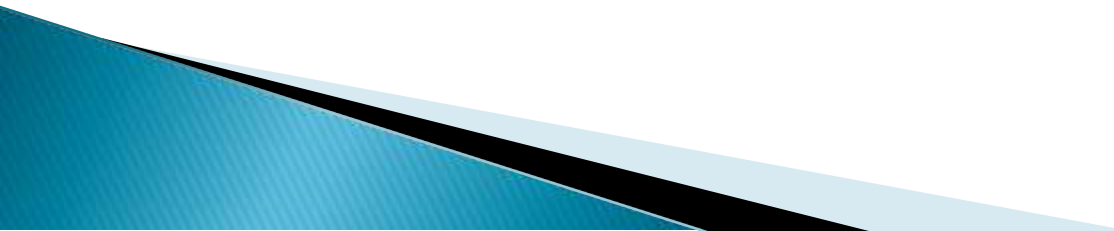
```
  {{ x.name + ', ' + x.country }}
```

```
</li>
```

```
</ul>
```

```
</div>
```

# Create New Directives

- ▶ In addition to all the built-in AngularJS directives, you can create your own directives.
  - ▶ New directives are created by using the `.directive` function.
  - ▶ To invoke the new directive, make an HTML element with the same tag name as the new directive.
  - ▶ When naming a directive, you must use a camel case name, `testDirective`, but when invoking it, you must use – separated name, `test-directive`
- 

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular
.min.js"></script>
<body ng-app="myApp">
```

```
<test-directive></w3-test-directive>
```

```
<script>
var app = angular.module("myApp", []);
app.directive("testDirective", function() {
  return {
    template : "<h1>Made by a directive!</h1>"
  };
});
</script>
```

```
</body>
```

```
</html>
```

## ▶ You can invoke a directive by using:

- Element name

```
<test-directive></test-directive>
```

- Attribute

```
<div test-directive></div>
```

- Class

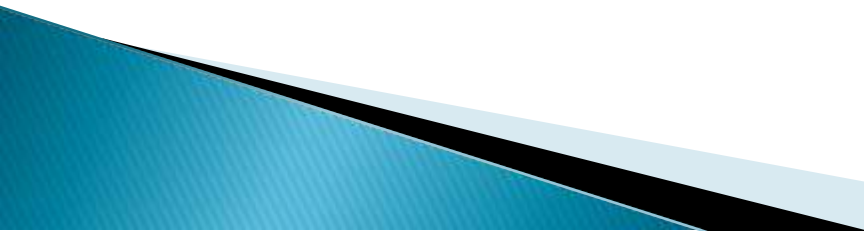
```
<div class="test-directive"></div>
```

- Comment

```
<!-- directive: test-directive -->
```



# Restrictions

- ▶ You can restrict your directives to only be invoked by some of the methods.
  - ▶ The legal restrict values are:
    - E for Element name
    - A for Attribute
    - C for Class
    - M for Comment
  - ▶ By default the value is EA, meaning that both Element names and attribute names can invoke the directive.
  - ▶
- 

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.
min.js"></script>
<body ng-app="myApp">
```

```
<test-directive></test-directive>
```

```
<div test-directive></div>
```

```
<script>
```

```
var app = angular.module("myApp", []);
app.directive("testDirective", function() {
  return {
    restrict : "A",
    template : "<h1>Made by a directive!</h1 >"
  };
}); </script>
```

<p><strong>Note:</strong> By setting the  
<strong>restrict</strong> property to "A", only the HTML element  
with the "test-directive" attribute has invoked the directive.</p>

```
</body></html>
```