GFT

ANGULARJS BUILT IN DIRECTIVES

Directives

GFT

CONCEPTS

1. ng-app
2. ng-model
3. ng-click
4. ng-class
5. ng-show / ng-hide / ng-disabled / ng-if
6. ng-switch
7. ng-repeat / $filters

ng-app

GFT

The **ng-app** directive defines an AngularJS application.

The ngApp directive designates the **root element** of the application and is typically placed near the root element of the page - e.g. on the <body> or <html> tags

```
<!DOCTYPE html>
<html ng-app="ExampleAPP">

<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
   <title data-ng-bind="$state.current.name"></title>
   <meta charset="utf-8">
   <meta name="viewport" content="initial-scale=1”>
    <link rel="stylesheet" href="styles/css/main.css">
</head>
```

index.html

NOTE:
If you need to validate your markup , you can express Angular directives as HTML5 data attributes. Just prefix data- to any angular attribute

ng-model

- ng-model is responsible for:
  - Binding the view into the model, which other directives such as input, textarea or select require.
  - Providing validation behavior (i.e. required, number, email, url).
  - Keeping the state of the control (valid/invalid, dirty/pristine, touched/untouched, validation errors).
  - Setting related css classes on the element (ng-valid, ng-invalid, ng-dirty, ng-pristine, ng-touched, ng-untouched, ng-empty, ng-not-empty) including animations.
  - Registering the control with its parent form.

```
<<input ng-model="example" />
{{example}}
```

NOTE:

ngModel will try to bind to the property given by evaluating the expression on the current scope. If the property doesn't already exist on this scope, it will be created implicitly and added to the scope

ng-click

GFT

- The ng-click directive defines an AngularJS click event.

html

```
<button ng-click="count = count + 1">Click Here!</button>
<p>{{ count }}</p>
```

javascript

```
angular
    .module('myApp')
    .controller('myCtrl', myCtrl);

    function myCtrl($scope){

    // ng-click will increment the counter
    $scope.count = 0;

    }
```

ng-class

- Using ng-class to change the color of text adding and removing css classes

html

```
<h3
 ng-class="{'text-danger':textButton , 'text-success': !textButton }" >
  NG-CLASS - Adding Conditional Classes to this text
</h3>
<button  class="btn btn-default"
            ng-model="textButton"
            ng-click="changeTextColor()">
             Change Color
</button>
```

javascript

```
angular
    .module('myApp')
    .controller('myCtrl', myCtrl);

    function myCtrl($scope){

    // Function to change true or false the scope variable
 $scope.textButton = false;

 $scope.changeTextColor = function(){
          $scope.textButton = !$scope.textButton;
        };

    }
```

ng-show / ng-hide / ng-disabled / ng-if

GFT

- Extending the last view example we will show and hide and disable conditionally some elements

html

```
<div class="well">
        <h3>Testing  ng-show / ng-hide / ng-disabled / ng-if</h3>
        <button  class="btn btn-info"
         ng-model="textButton"
         ng-click="changeTextColor()"> Test NG Directive</button>
         <br>
        <h5>Examples :</h5>
        <hr>
        <button class="btn btn-success" ng-disabled="textButton">
         NG-DISABLED
         </button>
        <h3 ng-hide="textButton">NG-HIDE</h3>
        <h3 ng-show="textButton">NG-SHOW</h3>
        <h3 ng-if="textButton">NG-IF</h3>

</div>
```

javascript

```
angular
    .module('myApp')
    .controller('myCtrl', myCtrl);

    function myCtrl($scope){

    // Function to change true or false the scope variable
 $scope.textButton = false;

 $scope.changeTextColor = function(){
          $scope.textButton = !$scope.textButton;
        };

    }
```

ng-switch

GFT

- Using ng-switch to change and show conditionally some elements

html

```
<div class="well">
        <h3>NG-SWITCH</h3>
       <input class="form-control"  ng-model="typeColor" >
       <div ng-switch on="typeColor">
          <h4>My color is :  </h4>
          <span ng-switch-when="red" class="text-danger" >I am Red </span>
          <span ng-switch-when="green" class="text-success" >I am Green</span>
          <span ng-switch-default>I am Default Text</span>
      </div>
    </div>
```

ng-class

- The ng-click directive defines an AngularJS click event.

html

```
<button ng-click="count = count + 1">Click Here!</button>
<p>{{ count }}</p>
```

javascript

```
angular
    .module('myApp')
    .controller('myCtrl', myCtrl);

    function myCtrl($scope){

    // ng-click will increment the counter
    $scope.count = 0;

    }
```

ng-repeat / $filter

GFT

- Create a loop to show elements from an array in the view and use filter and order to iterate

html

```
<div class="well">
      <h3>NG-REPEAT</h3>
      <input class="form-control" ng-model="search">
      <!--Create The ng-repeat-->
      <ul>
         <li ng-repeat="student in students |  orderBy: 'age' | filter:search
">{{student.name}} : {{student.age}}</li>
      </ul>
   </div>
```

javascript

```
angular
   .module('myApp')
   .controller('myCtrl', myCtrl);

   function myCtrl($scope){
         // List of Students
      $scope.students = [
       {
              name:'Jonh Doe',
              age:28,
              role:'Angular Developer',
              office: 'BCN',
              active:true
       },
               { //  … More Students  },
            ];
   }
```