**INTRODUCTION:**

AngularJS is a JavaScript framework. It is a library written in JavaScript.

AngularJS extends HTML with **ng-directives**.

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

The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.

The **ng-bind** directive binds application data to the HTML view.

The ng-bind directive tells AngularJS to replace the content of an HTML element with the value of a given variable, or expression. If the value of the given variable, or expression, changes, the content of the specified HTML element will be changed as well.

Eg:

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

<p>Total in dollar: <span ng-bind="quantity \* cost"></span></p>

</div>

The **ng-init** directive initializes AngularJS application variables.

Eg:

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

The **ng-change** Specifies an expression to evaluate when content is being changed by the user

Syntax: <element ng-change="expression"></element>

Supported by <input>, <select>, and  <textarea>.

Eg:

<input type="text" ng-change="myFunc()" ng-model="myValue" />

The **ng-repeat** directive repeats an HTML element, used on an array of objects

Eg:

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

{{ x }}

</li>

The **ng-class** Specifies CSS classes on HTML elements

Eg:

<select ng-model="home">

<option value="sky">Sky</option>

<option value="tomato">Tomato</option>

</select>

<div ng-class="home">

<h1>Welcome Home!</h1>

<p>I like it!</p>

</div>

AngularJS **modules** define AngularJS applications.

AngularJS **controllers** control AngularJS applications.

**ANGULARJS MODULES:**

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

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

# ANGULARJS CONTROLLERS:

AngularJS controllers **control the data** of AngularJS applications.

AngularJS controllers are regular **JavaScript Objects**.

# 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.

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

## AngularJS Filters:

AngularJS provides filters to transform data:

* currency Format a number to a currency format.
* date Format a date to a specified format.
* filter Select a subset of items from an array.
* json Format an object to a JSON string.
* limitTo Limits an array/string, into a specified number of elements/characters.
* lowercase Format a string to lower case.
* number Format a number to a string.
* orderBy Orders an array by an expression.
* uppercase Format a string to upper case.

Eg1: <div ng-app="myApp" ng-controller="personCtrl">

<p>The name is {{ lastName | uppercase }}</p>

</div>

Eg2: <li ng-repeat="x in names | orderBy:'country'">  
    {{ x.name + ', ' + x.country }}  
  </li>

Eg3: <h1>Price: {{ price | currency }}</h1>

# AngularJS Services

In AngularJS, a service is a function, or object, that is available for, and limited to, your AngularJS application.

To create your own service, connect your service to the module.

Once you have created a service, and connected it to your application, you can use the service in any controller, directive, filter, or even inside other services.

# AngularJS AJAX - $http

**$http** is an AngularJS service for reading data from remote servers. The AngularJS $http service makes a request to the server, and returns a response.

## Methods

* .delete()
* .get()
* .head()
* .jsonp()
* .patch()
* .post()
* .put()

The methods above are all shortcuts of calling the $http service

## Properties

The response from the server is an object with these properties:

* .config the object used to generate the request.
* .data a string, or an object, carrying the response from the server.
* .headers a function to use to get header information.
* .status a number defining the HTTP status.
* .statusText a string defining the HTTP status.

To handle errors, add one more functions to the .then method.

## JSON

The data you get from the response is expected to be in JSON format.

JSON is a great way of transporting data, and it is easy to use within AngularJS, or any other JavaScript.

# AngularJS Tables

The ng-repeat directive is perfect for displaying tables.

Eg:

<table>

<tr ng-repeat="x in names">

<td>{{ x.Name }}</td>

<td>{{ x.Country }}</td>

</tr>

</table>

To display uppercase, add an **uppercase** filter

<table>  
  <tr ng-repeat="x in names">  
    <td>{{ x.Name }}</td>  
    <td>{{ x.Country | uppercase }}</td>  
  </tr>  
</table>

# Angular DOM

AngularJS has directives for binding application data to the attributes of HTML DOM elements.

The **ng-disabled** directive binds AngularJS application data to the disabled attribute of HTML elements.

<button ng-disabled="mySwitch">Click Me!</button>

The **ng-show** directive shows or hides an HTML element.

<p ng-show="true">I am visible.</p>  
  
<p ng-show="false">I am not visible.</p>

The ng-repeat directive has its limitations, the selected value must be a string.

When using the ng-options directive, the selected value can be an object.