ANGULARJS

by Google

This PPT covers below topics

- **Introduction and definition of AngularJS.**
- **Some of the key points about AngularJS.**
- **Basic example.**
- **The flow of the angular page.**
- **What is Directive ? Some examples.**
- **Understanding Controllers ? Examples.**
- **How to create module?.**
- **Creating controller in module.**
- **Template-expanding directive Example.**
- **Filters and Examples.**

Introduction

- Client side programming is becoming the one of the main parts of web applications. Now a days, We are seeing the explosion of JavaScript libraries. And the reason is that some JavaScript libraries got very popular and developers took them hand to hand because of its cool feature and almost negligible performance cost, Now in our projects, count of JavaScript files (plugins and custom files) are increasing rapidly which is making it unmanageable and unmaintainable**.** AngularJS provides best of the both the worlds and now a days, it is one of the most talked and used JavaScript framework in web applications.

What is AngularJS?

- **AngularJS is not just another JavaScript library but it provides a framework to write a proper architectured, maintainable and testable client side code.**

Continue…

Some of the key points are…

- It follows MVC framework. If you don’t have Idea MVC framework, I’ll suggest you to get an Idea of MVC framework and then start learning AngularJS.

- AngularJS is primarily aimed to develop SPAs (Single Page Applications), it means your single HTML document turns into application. But it is also used a lot in ASP.NET and other applications .

- Allows you to write Unit and integration tests for JavaScript code. Testability is one of the main points, which was kept in mind while writing this framework so it has great support of it

- It provides it’s own and very rich list of attribute and properties for HTML controls which increases the usability of it exponentially. It is also called directives.

Continue…

Some of the key points are…

- Supports two-way binding that means your data model and control's data will be in sync.

- Angular library is also available on CDN so you just need to the url of the CDN and it available for use.

Note: A content delivery network or content distribution network (CDN) is a large distributed system of servers deployed in multiple data centers across the Internet.

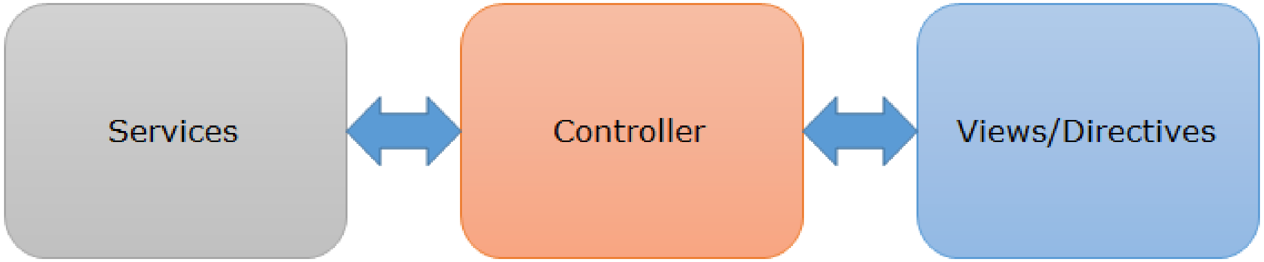
- It is an Open Source
  - AngularJS is a open source library and developed and supported by Google. Being an open source, you can go through the code itself and customize it if required. There is lot of support from JavaScript community and even you can contribute to it. Currently, more that 100 contributors have contributed and it is increasing day by day.

Continue…

So let’s discuss the main components of AngularJS.

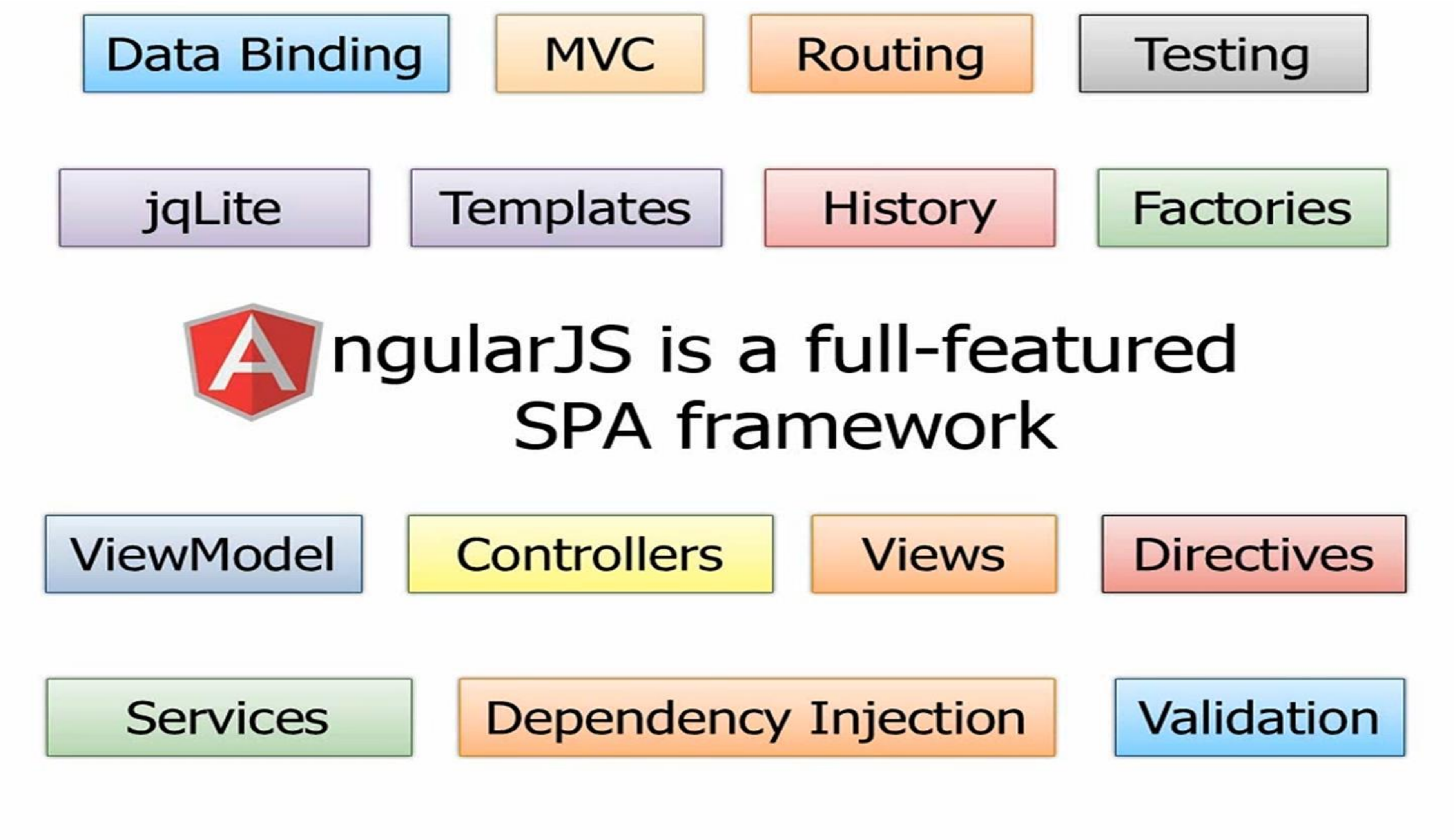
- **Controller – It is main component of AngularJS and contains the state and logic both. It acts as bridge between services and views.**
- **Views/Directives – Here we generate the UI. Directives extends the HTML element and enables us to generate the complex html easily. Controllers talks to view to both directions.**
- **Services – It contains the core logic and state of the application. In it, we can communicate to server to get and post the data.**

- **See all the above components glued with each other**



Continue…

Some of the features are…



Data Binding

MVC

Routing

Testing

jqLite

Templates

History

Factories

AngularJS is a full-featured SPA framework

ViewModel

Controllers

Views

Directives

Services

Dependency Injection

Validation

Getting Started…..

- To use AngularJS, you must include it as a reference in your HTML page

```
<script src=“angular.min.js”></script>
```

(OR)

```
<script src="http://code.angularjs.org/angular-1.0.1.js"></script>
```

(OR)

```
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular
.min.js" type="text/javascript"></script>
```

Note: Use Google’s CDN is recommended. Google’s servers are very fast, and the scripts cacheable across applications.

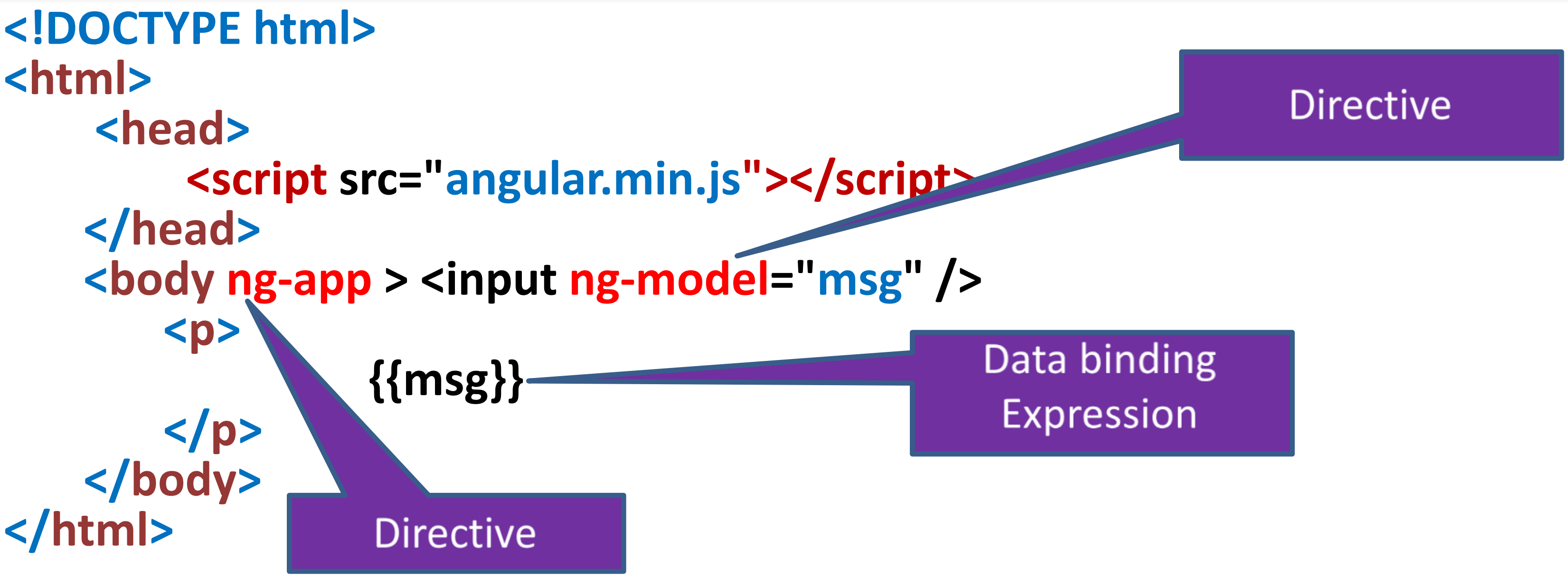
- Add a directive in HTML tag called ng-app to indicate that this is using Angular JS.

`<html ng-app></html>` (or) `<body ng-app></body>`

Directive

Directive

Basic Example of AngularJS



```
<!DOCTYPE html>
<html>
    <head>
        <script src="angular.min.js"></script>
    </head>
    <body ng-app > <input ng-model="msg" />
        <p>
            {{msg}}
        </p>
    </body>
</html>
```

Directive

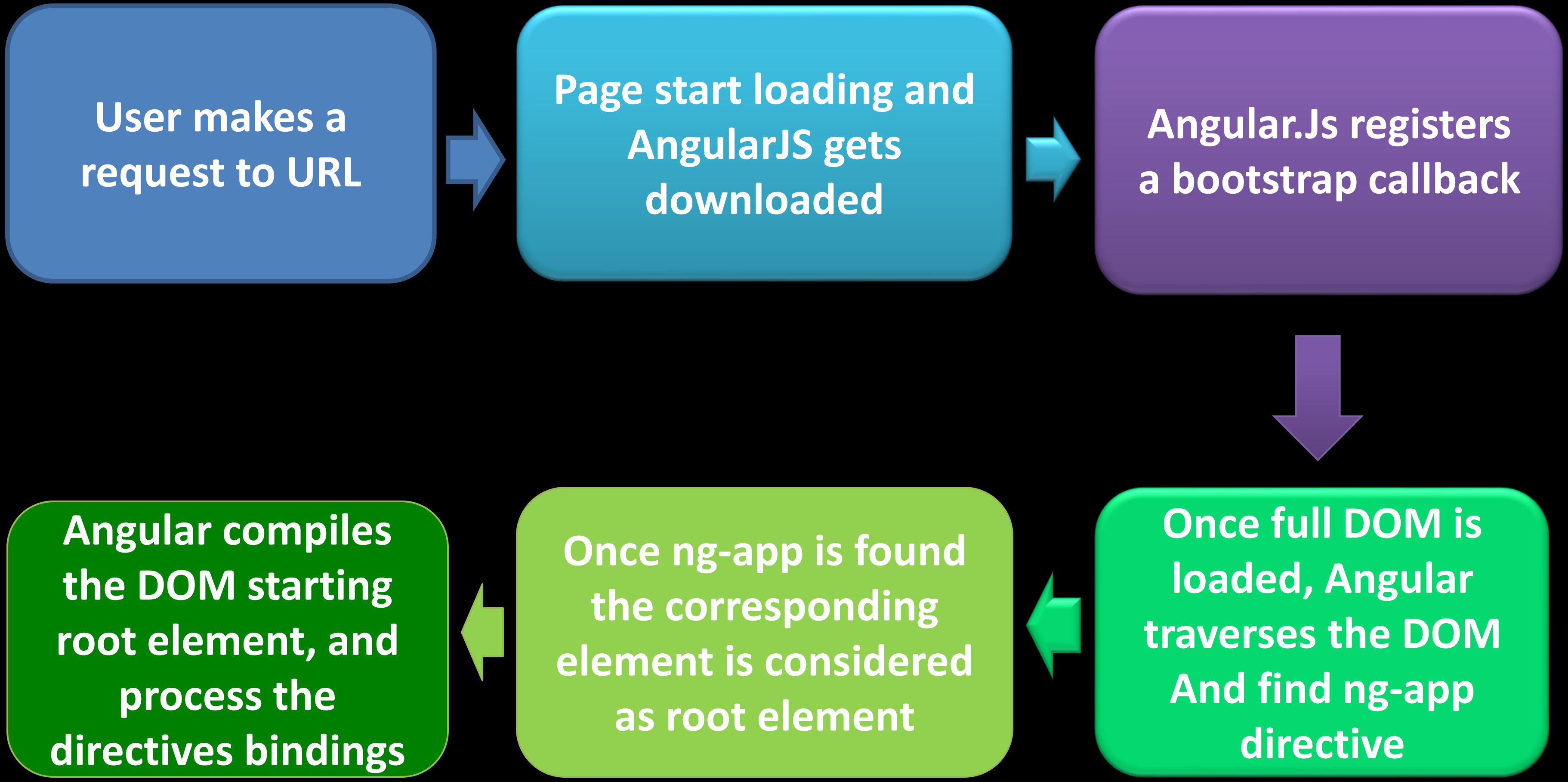
Data binding Expression

Directive

When AngularJS loads, it scans the document for the ng-app attribute. This tag is usually set to the name of the application's main module. Once the ng-app attribute is found, Angular will process the document, loading the main module and its dependencies, scanning the document for custom directives, and so on.

The ngModel directive binds an input,select, textarea (or custom form control) to a property on the scope using NgModelController, which is created and exposed by this directive.

The flow of the angular page



User makes a request to URL

Page start loading and AngularJS gets downloaded

Angular.Js registers a bootstrap callback

Once full DOM is loaded, Angular traverses the DOM And find ng-app directive

Once ng-app is found the corresponding element is considered as root element

Angular compiles the DOM starting root element, and process the directives bindings

Directives

- It provides it's own and very rich list of attribute and properties for HTML controls which increases the usability of it exponentially. It is also called directives.

- Directives are most powerful feature of Angular Js and they will let you specify how your page should be structured for the data available in a given scope.

- There are 50+ directives in angular Js which we are using in many ways.
- Here all words prefixing with "ng" keyword are directives.

- Some useful directives are ng-repeat , ng - init, ng-show , ng-hide, ng-model, ng- switch ,ng – if, ng-model-instant , ng-controller ,ng-click etc….

- Some time it is prefixed with data like "data - ng- switch". Both are of same meaning.

Directives

At a high level, directives are markers on a DOM element (such as an attribute, element name, or CSS class) that tell AngularJS's HTML compiler ($compile) to attach a specified behavior to that DOM element or even transform the DOM element and its children

a
form
input
input.checkbox
input.email
input.number
input.radio
input.text
input.url
ngApp
ngBind
ngBindHtml
ngBindTemplate

ngBlur
ngChange
ngChecked
ngClass
ngClassEven
ngClassOdd
ngClick
ngCloak
ngController
ngCopy
ngCsp
ngCut
ngDblclick

ngDisabled
ngFocus
ngForm
ngHide
ngHref
ngIf
ngInclude
ngInit
ngKeydown
ngKeypress
ngKeyup
ngList
ngModel

ngMouseleave
ngMousemove
ngMouseover
ngMouseup
ngNonBindable
ngOpen
ngPaste
ngPluralize
ngReadonly
ngRepeat
ngSelected
ngShow
ngSrc

ngSrc
ngSrcset
ngStyle
ngSubmit
ngSwitch
ngTransclude
ngValue
script
select
textarea

Example of Directive

```
<!DOCTYPE html>
<html>
     <head>
          <script src="angular.min.js"></script>
    </head>
    <body ng-app ng-init="msg = 'hello world'"> <input ng-model="msg"
    />
         <p>
                    {{msg}}
         </p>
    </body>
</html>
```

Directive

Directive

In this example, the ng-init attribute initializes an msg variable to "hello world" and the ng-model attribute binds the content of the variable to an input element. The text enclosed in curly braces is a binding expression. AngularJS evaluates the expression and updates the document whenever the value of the expression changes.

Understanding Controllers

- In Angular, a Controller is a JavaScript constructor function that is used to augment the Angular Scope.

- The ngController directive attaches a controller class to the view

- When a Controller is attached to the DOM via the ng-controller directive, Angular will instantiate a new Controller object, using the specified Controller's constructor function. A new child scope will be available as an injectable parameter to the Controller's constructor function as $scope.

Use controllers to:

- Set up the initial state of the $scope object.
- Add behavior to the $scope object.

Continue…

Example of Controller in module

```
<!DOCTYPE html>
    <html ng-app="nameApp">
    <head>
        <title>:: Angular JS Example</title>
        <script src="angular.min.js"></script>
        <script>
              var nameApp=angular.module('nameApp',[]);
              nameApp.controller('NameCtrl',function($scope){
              $scope.firstName='Rambabu';
              $scope.lastName='Ambala';
              });
        </script>

     </head>

    <body ng-controller="NameCtrl">
         First Name: <input  ng-model="firstName" type="text" /><br>
         Last Name: <input  ng-model="lastName" type="text" />
         Hello {{firstName}} {{lastName}}
     </body>
    </html>
```

Example of “ng-repeat” Directive

```
<!DOCTYPE html>
<html ng-app="nameApp">
<head>
		<title>:: Angular JS Example</title>
		<script src="angular.min.js"></script>
		<script>
				var nameApp=angular.module('nameApp',[]);
				nameApp.controller('NameCtrl',function($scope){
				$scope.names=['Rambabu','Devish','Ambala'];
				});
		</script>
 </head>
<body ng-controller="NameCtrl">
		<ul>
				<li ng-repeat="name in names">
						{{name}}
				</li>
		</ul>
 </body>
</html>
```

Directive

Directive

Filters

- **Selects a subset of items from array and returns it as a new array.**
- **Filters does the formatting of output as per user requirement.**
- **This can lead to features such as searching, ordering, and more.**

Types of Filters

- Currency
- date
- filter
- json
- limitTo
- lowercase
- number
- orderBy
- uppercase

Example of Filter

Template-expanding directive.html

```
<!DOCTYPE html>
<html>
    <head>
   <title>:: Angular JS Example</title>
   <script rc="angular.min.js"></script>
   <script src="ExpandingDerective.js"></script>
    </head>
    <body ng-controller="CountryCtrl">
Search : <input ng-model="query"
type="text">
<table >
            <tr><th>Country</th>
            <th>Population</th>
</tr>
<tr ng-repeat="country in countries |
          filter:query | orderBy: '-population'">
            <td>{{country.name}}</td>
            <td>{{country.population}}</td>
</tr>
</table>
 </body></html>
```

ExpandingDerective.js

```
var countryApp=angular.module('countryApp',[]);
countryApp.controller('CountryCtrl',function($scope){
 $scope.countries=[
{"name": "China", "population": 1365370000},

{"name": "India", "population": 1246160000},

{"name": "United States", "population": 318320000},

{"name": "Indonesia", "population": 252164800},

{"name": "Brazil", "population": 202794000},

{"name": "Pakistan", "population": 188020000},

{"name": "Nigeria", "population": 178517000  },

{"name": "Bangladesh", "population": 156557000},

{"name": "Russia", "population": 146000000}];
```

Angular X

- From AngularJS to Angular X
- Though AngularJS introduced many features to develop single-page applications, it started to stay behind as and when new advancements in javascript emerged. For example, the elimination of a colossal bundle size when compared to other libraries, which leads to performance issues. One of Google's competitors introduced ReactJS with better performance than angular, and that highlighted the drawbacks of AngularJS. Therefore, Google had to rewrite the framework altogether due to imperfections. That is how Angular 2 has come into existence.

- Angular is Type Script Based.

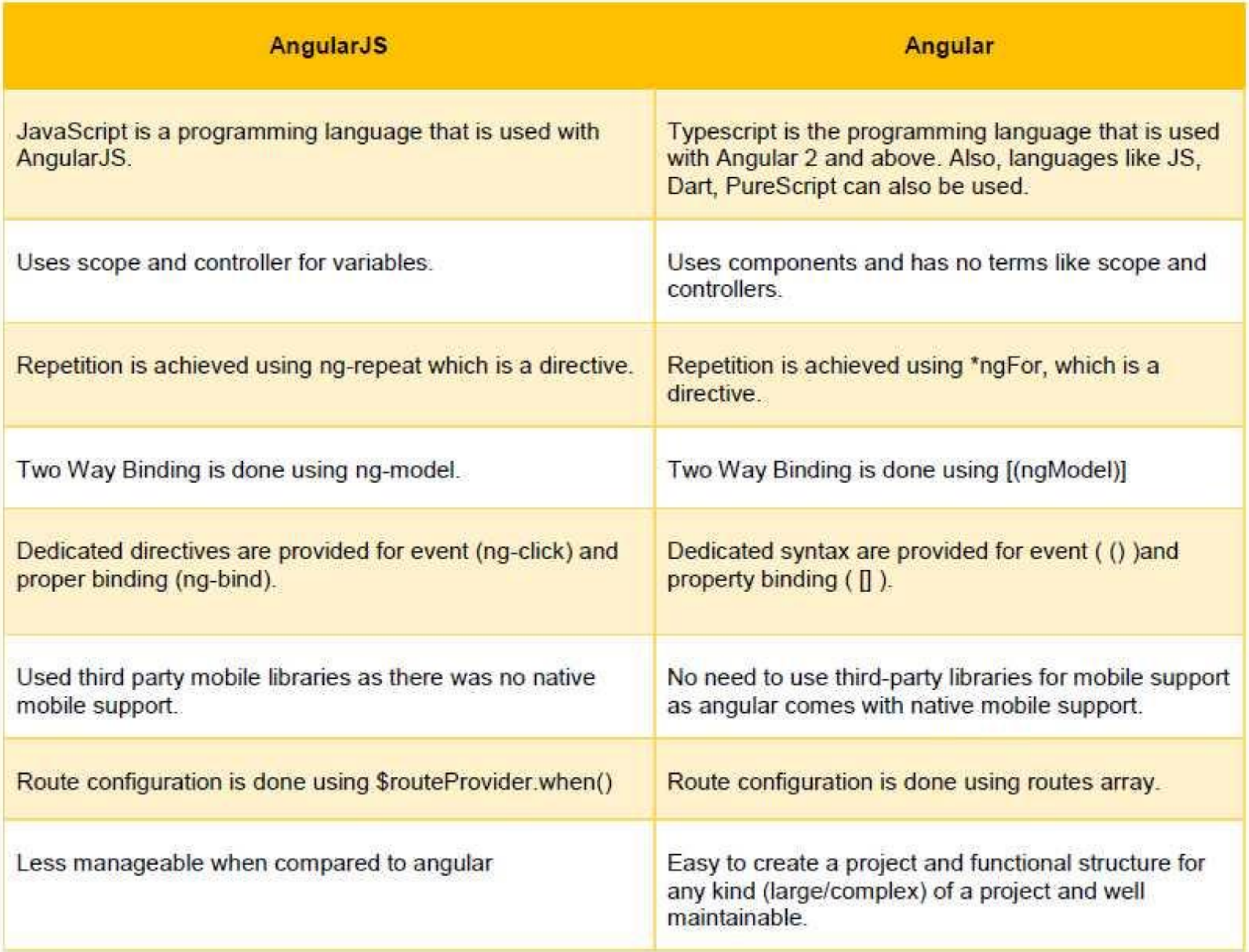
- Many companies are using angular X as it benefits the developers for many reasons such as:

- AOT (Ahead-of-Time) Compilation which converts the HTML and Typescript code into Javascript during the build time. Advantage of AOT compilation is that it compiles the code first and then the browser loads the compiled code which results in faster rendering.

- Angular CLI (stands for Command Line Interface) is a command-line tool that helps in creating and serving angular applications. Once a project is designed for further development, you will have to to create and configure the building blocks such as components, pipes, services, directives. Rest of the things are taken care of by the cli. CLI also takes care of the build of a project and ultimately makes the project development and testing much more comfortable.

- Component-based architecture really helps an application to divide into logical and functional components. Components are independent of each other, which also makes testing easy.

- HTML elements can act dynamically by using directives.



AngularJS	Angular
JavaScript is a programming language that is used with AngularJS.	Typescript is the programming language that is used with Angular 2 and above. Also, languages like JS, Dart, PureScript can also be used.
Uses scope and controller for variables.	Uses components and has no terms like scope and controllers.
Repetition is achieved using ng-repeat which is a directive.	Repetition is achieved using *ngFor, which is a directive.
Two Way Binding is done using ng-model.	Two Way Binding is done using [(ngModel)]
Dedicated directives are provided for event (ng-click) and proper binding (ng-bind).	Dedicated syntax are provided for event (())and property binding ([]).
Used third party mobile libraries as there was no native mobile support.	No need to use third-party libraries for mobile support as angular comes with native mobile support.
Route configuration is done using $routeProvider.when()	Route configuration is done using routes array.
Less manageable when compared to angular	Easy to create a project and functional structure for any kind (large/complex) of a project and well maintainable.

What is Nx?

- Angular CLI power-ups for **modern development** maintained by Nrwl helps manage workspaces in a ***monorepo*** way and let us develop like a Google.
- Extends the Angular CLI to provide useful tools that make **full-stack development** much easier.

- Why Nx?
- Nrwl helps the Fortune 500 build Angular applications.
- **These companies don't build small apps.** They have multiple teams building multiple apps using multiple shared libs. It's many to many to many. Organizing this dev workflow is challenging.
- **They care about consistency.** If every team uses their own unique way of building software, the code is harder to reuse and integrate.
- **They have legacy AngularJS apps they need to upgrade.** NgUpgrade is great, but it is easy to misconfigure.
- **They want to write robust, proven code**: error handling, race conditions, etc.

- State Management
- **Managing state and side effects is probably the hardest problem in frontend development.**
- NgRx is the de facto state-management library used in the Angular community. It gives you a lot of freedom, but it means that it leaves a lot up to you to figure out. As a result, teams spend time debating how it should be used, and different teams come up with their own incompatible solutions. Not only is it an inefficient use of developer time, it also inhibits teams' abilities to share knowledge and tools.

THANK
YOU