

1. Why AngularJS?

Ans: HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications.

1. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.
2. Two way data-binding, form validations, routing supports, inbuilt services.
3. REST friendly.
4. Dependency injection support.
5. It helps you to structure and test your JavaScript code.

2. Is AngularJS a library or Framework?

Ans: AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and let you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.

3. What is difference between library and Framework?

Ans:

Library - It is a collection of objects/functions/methods (depending on your language) and your application "links" against it and thus can use the objects/functions/methods. It is basically a file containing re-usable code that can usually be shared among multiple applications (you don't have to write the same code over and over again).

Framework - A framework is a re-usable design for a software system (or subsystem). A software framework may include support programs, code libraries, a scripting language, or other software to help develop and glue together the different components of a software project. Various parts of the framework may be exposed through an API.

A framework can be a group of libraries, group of scripts, group of languages which supports for applications that can be developed using an environment.

4. Can we use Angular JS with JQuery?

Ans: Yes, but that's a bad practice to use jQuery when we stepped into angular once. We need to stay away from jQuery and try to realize the solution with an AngularJS approach, where there is solution for every approach that we used in JQuery.

If the user wants to use Angular with JQuery, then the problem is that in most cases that jQuery plugin could be rewritten in AngularJS in a fraction of the code, where suddenly everything becomes comprehensible and straightforward.

5. What are the advantages of using AngularJS?

Ans:

1. AngularJS supports MVC pattern

2. Can do two ways data binding using AngularJS
3. It has per-defined form validations
4. It supports both client server communication
5. Can develop SPA (single page applications)
6. Modular development
7. Increase testability
7. Can add custom directives.
8. Supports Dependency Injection.
9. Event Handlers.

6. What does the 'ng' stands for in angular directives?

Ans: The prefix ng stands for "Angular" all of the built-in directives that ship with Angular use that prefix. Similarly, it is recommended that you do not use the ng prefix on your own/custom directives in order to avoid possible name collisions in future versions of Angular. And also that 'ng' sounds like Angular.

7. What is the importance of Angular Models ?

Ans: The ngModel directive binds an input, select, textarea (or custom form control) to a property on the scope. For implementing Angular Models ng-model/data-ng-model directive should be used, where ng-model provides the facility of two-way binding. Two-way binding defines that the data can be passed from controller to view using scope and vice versa.

ngModel is responsible for:

1. Binding the view into the model, which other directives such as input, textarea or select require.
2. Providing validation behavior (i.e. required, number, email, url).
3. Keeping the state of the control (valid/invalid, dirty/pristine, touched/untouched, validation errors).
4. 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.
5. Registering the control with its parent form.

8. Can we use multiple controllers on one page ?

Ans: Yes, absolutely there is no problem in declaring multiple controllers in a single page application.

Every time when DOM finds a ng-controller directive it creates a new instance for that controller.

AngularJS does not support only multiple ng-app directives on one HTML page.

Example:

```
<body ng-app="myApp">
  <div ng-controller="myCtrl1">
    <p>Stuff here</p>
  </div>
  <div ng-controller="myCtrl2">
```

```
<p>Other stuff here</p>
</div>
</body>
<script>
var app = angular.module('myApp',[]);
app.controller('myCtrl1',function($scope){
});
app.controller('myCtrl2',function($scope){
});
</script>
```

9. Is AngularJS is compatible with all modern browsers?

Ans: YES, AngularJS team run extensive test suite against the following browsers: Safari, Chrome, Firefox, Opera 15, IE9 and mobile browsers (Android, Chrome Mobile, iOS Safari).

References:

<https://docs.angularjs.org>

www.quora.com