1. What is Angular.js?

AngularJS is open source client side MV* (Model – View – Whatever) framework for creating dynamic web applications. It gives life to your static HTML and makes it dynamic with its magic. It extends HTML using directives, expression and data binding techniques to define a powerful HTML template.

2. Explain what are the key features of Angular.js ?

The key features of angular.js are

Scope
Controller
Model
View
Services
Data Binding
Directives
Filters
Testable

3. Explain what is scope in Angular.js ?

Scope refers to the application model, it acts like glue between application controller and the view. Scopes are arranged in hierarchical structure and impersonate the DOM (Document Object Model) structure of the application. It can watch expressions and propagate events.

4. **Explain what is services in Angular.js ?**

In angular.js services are the singleton objects or functions that are used for carrying out specific tasks. It holds some business logic and these function can be called as controllers, directive, filters and so on.

5. **Explain what is Angular Expression? Explain what is key difference between angular expressions and JavaScript expressions?**

Like JavaScript, Angular expressions are code snippets that are usually placed in binding such as {{ expression }}

The key difference between the JavaScript expressions and Angular expressions

Context : In Angular, the expressions are evaluated against a scope object, while the Javascript expressions are evaluated against the global window

Forgiving: In Angular expression evaluation is forgiving to null and undefined, while in Javascript undefined properties generates TypeError or ReferenceError

No Control Flow Statements: Loops, conditionals or exceptions cannot be used in an angular expression

Filters: To format data before displaying it you can use filters

6. **With options on page load how you can initialize a select box ?**

You can initialize a select box with options on page load by using ng-init directive

<div ng-controller = " apps/dashboard/account " ng-switch
On = "! ! accounts" ng-init = " loadData ( ) ">

7. **Explain what are directives ? Mention some of the most commonly used directives in Angular.js application ?**

A directive is something that introduces new syntax, they are like markers on DOM element which attaches a special behavior to it. In any Angular.js application, directives are the most important

components.
Some of the commonly used directives are ng-model, ng-App, ng-bind, ng-repeat , ng-show etc.

8. **Mention what are the advantages of using Angular.js ?**
Angular.js has several advantages in web development.

Angular.js supports MVS pattern
Can do two ways data binding using Angular.js
It has per-defined form validations
It supports both client server communication
It supports animations

9. Explain what Angular JS routes does ?
Angular js routes enable you to create different URLs for different content in your application. Different URLs for different content enables user to bookmark URLs to specific content. Each such bookmarkable URL in Angular.js is called a route
A value in Angular JS is a simple object. It can be a number, string or JavaScript object. Values are typically used as configuration injected into factories, services or controllers. A value should be belong to an Angular.js module.
Injecting a value into an Angular.js controller function is done by adding a parameter with the same name as the value

10. Explain what is data binding in Angular.js ?
Automatic synchronization of data between the model and view components is referred as data binding in Angular.js. There are two ways for data binding

Data mining in classical template systems
Data binding in angular templates

11. What makes angular.js better ?
Registering Callbacks: There is no need to register callbacks . This makes your code simple and easy to debug.
Control HTML DOM programmatically: All the application that are created using Angular never have to manipulate the DOM although it can be done if it is required.
Transfer data to and from the UI: Angular.js helps to eliminate almost all of the boiler plate like validating the form, displaying validation errors, returning to an internal model and so on which occurs due to flow of marshalling data.
No initilization code: With angular.js you can bootstrap your app easily using services, which auto-injected into your application in Guice like dependency injection style.

12. Is AngularJS extensible?
Yes! In AngularJS we can create custom directive to extend AngularJS existing functionalities. Custom directives are used in AngularJS to extend the functionality of HTML. Custom directives are defined using "directive" function. A custom directive simply replaces the element for which it is activated. AngularJS application during bootstrap finds the matching elements and do one time activity using its compile() method of the custom directive then process the element using link() method of the custom directive based on the scope of the directive.

13. Explain what is string interpolation in angular.js ?
In angular.js the compiler during the compilation process matches text and attributes using interpolate service to see if they contains embedded expressions. As part of normal digest cycle these expressions are updated and registered as watches.

14. Mention the steps for the compilation process of HTML happens?

Compilation of HTML process occurs in following ways

Using the standard browser API, first the HTML is parsed into DOM

By using the call to the $compile () method, compilation of the DOM is performed. The method traverses the DOM and matches the directives.

Link the template with scope by calling the linking function returned from the previous step

15. Explain what is directive and Mention what are the different types of Directive?

During compilation process when specific HTML constructs are encountered a behaviour or function is triggered, this function is referred as directive. It is executed when the compiler encounters it in the DOM.

Different types of directives are

Element directives

Attribute directives

CSS class directives

Comment directives

16. Explain what is linking function and type of linking function?

Link combines the directives with a scope and produce a live view. For registering DOM listeners as well as updating the DOM, link function is responsible. After the template is cloned it is executed.

Pre-linking function: Pre-linking function is executed before the child elements are linked. It is not considered as the safe way for DOM transformation.

Post linking function: Post linking function is executed after the child elements are linked. It is safe to do DOM transformation by post-linking function

17. Explain what is injector?

An injector is a service locator. It is used to retrieve object instances as defined by provider, instantiate types, invoke methods and load modules. There is a single injector per Angular application, it helps to look up an object instance by its name.

18. Explain what is the difference between link and compile in angular.js?

Compile function: It is used for template DOM Manipulation and collect all of the directives.

Link function: It is used for registering DOM listeners as well as instance DOM manipulation. It is executed once the template has been cloned.

19. Explain what is factory method in angular.js?

For creating the directive, factory method is used. It is invoked only once, when compiler matches the directive for the first time. By using $injector.invoke the factory method is invoked.

20. Mention what are the styling form that ngModel adds to CSS classes ?

ngModel adds these CSS classes to allow styling of form as well as control

ng- valid

ng- invalid

ng-pristine

ng-dirty

21. Mention what are the characteristics of “Scope”?

To observer model mutations scopes provide APIs ($watch)

To propagate any model changes through the system into the view from outside of the Angular

realm
A scope inherits properties from its parent scope, while providing access to shared model properties, scopes can be nested to isolate application components
Scope provides context against which expressions are evaluated

22. Explain what is DI (Dependency Injection) and how an object or function can get a hold of its dependencies ?

DI or Dependency Injection is a software design pattern that deals with how code gets hold of its dependencies. In order to retrieve elements of the application which is required to be configured when module gets loaded , the operation “config” uses dependency injection.
These are the ways that object uses to hold of its dependencies
Typically using the new operator, dependency can be created
By referring to a global variable, dependency can be looked up
Dependency can be passed into where it is required

23. Mention what are the advantages of using Angular.js framework ?

Advantages of using Angular.js as framework are
Supports two way data-binding
Supports MVC pattern
Support static template and angular template
Can add custom directive
Supports REST full services
Supports form validations
Support both client and server communication
Support dependency injection
Applying Animations
Event Handlers

24. Explain the concept of scope hierarchy? How many scope can an application have?

Each angular application consist of one root scope but may have several child scopes. As child controllers and some directives create new child scopes, application can have multiple scopes. When new scopes are formed or created they are added as a children of their parent scope. Similar to DOM, they also creates a hierarchical structure.

25. Explain what is the difference between angular.js and backbone.js?

Angular.js combines the functionalities of most of the 3rd party libraries, it supports individual functionalities required to develop HTML5 Apps. While Backbone.js do their jobs individually.

26. Who created Angular JS ?

Intially it was developed by Misko Hevery and Adam Abrons. Currently it is being developed by Google.

27. What is AngularJS?

AngularJS is open source client side MV* (Model – View – Whatever) framework for creating

dynamic web applications. It gives life to your static HTML and makes it dynamic with its magic. It extends HTML using directives, expression and data binding techniques to define a powerful HTML template.

28. Is AngularJS a framework, library or a plugin?

The suitable answer is that its a framework. As its lightweight so people also get confused between library and framework.AngularJS is open source client side MVC framework for creating dynamic web applications.

29. Is it same as jQuery?

NO. jQuery is great library for manipulating the DOM, providing better user experience with animations and effects. You can create website using jQuery but not a web application. jQuery is just a library to play around with HTML, where as AngularJS is a framework to build a dynamic web app as it supports two data binding, MVC, allow testability, templates and many more. Its like AngularJS like a toolbox and jQuery is just a tool. You can read more here.

30. How can we decrease digest cycle time ?

To decrease digest cycle time you need to decrease the number of watchers. Below are some best practices you can follow to decrease number of watchers :-

Remove unnecessary watchers.
Use one time Angular binding. Especially if you see ng-repeat loop apply one time binding.
Work in batches.
Cache DOM
Use Web worker

31. Does Angular use the jQuery library?

YES, Angular can use jQuery if it's present in your app when the application is being bootstrapped. If jQuery is not present in your script path, Angular falls back to its own implementation of the subset of jQuery that we call jQLite.

32. Why AngularJS?

AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.
MVC implementation is done right.
It extends HTML using directives, expression and data binding techniques to define a powerful HTML template.
Two way data-binding, form validations, routing supports, inbuilt services.
REST friendly.
Dependency injection support.
It helps you to structure and test your JavaScript code.

33. What are the key features/concepts of Angular.js?

When you start learning AngularJS, you will come across following terms and these are the

features/concept of AngularJS.
Scope
Directives
Expression
Filters
Data Bindings
Model
View
Controller
Modules
Services
Dependency Injection

34. Is AngularJS is compatible with all modern browsers?

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).

35. What is the basic need to start with AngularJS?

To start with AngularJS, one need to make reference of angular.js. The latest version of AngularJS can be downloaded from AngularJS.com. You can either download the required js file and then host them locally or you can also use google CDN for referencing it. Here is the link for google CDN url for referencing AngularJS.

36. How does the AngularJS framework initialize itself?

The AngularJS has a self-executing anonymous function present in angular.js code, which does the initialization of AngularJS. Here is how below it looks,

```
(function(window, document, undefined) {
<!--
here goes entire AngularJS code
including functions, services, providers etc related code goes here
-->
if (window.angular.bootstrap) {
  //AngularJS is already loaded, so we can return here...
 console.log('WARNING: Tried to load angular more than once.');
 return;
 }
 //try to bind to jquery now so that one can write angular.element().read()
  //but we will rebind on bootstrap again.
 bindJQuery();
 publishExternalAPI(angular);
 jqLite(document).ready(function() {
  angularInit(document, bootstrap);
 });
})(window, document);
```

Above function first check if Angular has already been setup. If it has, we return here to prevent Angular from trying to initialize itself a second time. And then it binds jQuery (if present) and publish external API. And finally angularInit() method does the trick for initialization of AngularJS.

Following articles are recommended to know in detail.
How AngularJS Works – Explained with Angular Code
Dissecting Angular: Initializing An App

37. What is the bootstrapping in AngularJS?

Bootstrapping in AngularJS is nothing but just initializing, or starting, your Angular app. AngularJS supports automatic bootstrapping as well as manual way as well.

38. What are templates in AngularJS?

In Angular, templates are written with HTML that contains Angular-specific elements and attributes. Angular combines the template with information from the model and controller to render the dynamic view that a user sees in the browser. In other words, if your HTML page is having some Angular specific elements/attributes it becomes a template in AngularJS.

39. What are directives in AngularJS?

Directives are markers on a DOM element (such as an attribute, element name, comment or CSS class) that tell AngularJS to attach a specified behavior to that DOM element or even transform the DOM element and its children. When AngularJS finds the directive at the time of rendering then it attaches the requested behavior to the DOM element. Angular comes with a set of these directives built-in, like ngBind, ngModel, and ngClass.

40. Can we create our own directives?

YES. AngularJS allows us to create our own custom directive.

41. What are different type or classification of directives?

AngularJS directives can be classified in 4 different types .
Element directives
<ng-directive></ng-directive>
Attribute directives
<span ng-directive>
CSS class directives

Comment directives
<!-- directive: ng-directive expression -->

42. What is the name of directive is used to bootstrap an angular app?

ng-app directive is used to auto-bootstrap an AngularJS application. The ng-app directive defines the root element of the application and is typically present in the root element of the page - e.g. on the <body> or <html> tags.

43. Can AngularJS have multiple ng-app directives in a single page?

The answer is NO. Only one AngularJS application can be auto-bootstrapped per HTML document. The first ngApp found in the document will be used to define the root element to auto-bootstrap as an application. If you have placed another ng-app directive then it will not be processed by AngularJS. You need to manually bootstrap the second app, instead of using second ng-app directive. Read

44. Can angular applications (ng-app) be nested within each other?

NO. AngularJS applications cannot be nested within each other.

45. Can you bootstrap multiple angular applications on same element?

NO. If you try to do that then it will show an error "App Already Bootstrapped with this Element". This usually happens when you accidentally use both ng-app and angular.bootstrap to bootstrap an application. You can also get this error if you accidentally load AngularJS itself more than once.

46. In how many different ways, you can define a directive and what is the best practice?

Angular generally prefers camelCase for directives. But since HTML is not case-sensitive so it refers to directive in DOM in lower case form, delimited by dash (eg. ng-app). But when Angular complies then it normalize the directives.

Below are example of valid directive declaration.

ng-model
ngModel
ng:model
ng_model
data-ng-model
x-ng-model

The normalization process is as follows:

1. Strip x- and data- from the front of the element/attributes.
2. Convert the :, -, or _-delimited name to camelCase.

The best practice to use dash-delimited (ng-model) or directly camelCase form (ngModel). If you are using HTML validation tool, then it is advised to use data- prefixed version. And it also answers another question which is "Difference between ng-* and data-ng-*".

47. Mention some angularJS directives and their purpose?

The beauty of AngularJS directives is that they are self explainatory. By just looking at directive name, you will get the idea about purpose and use of directive. Below are some mostly used directives.

ng-app : Initializes application.
ng-model : Binds HTML controls to application data.
ng-Controller : Attaches a controller class to view.
ng-repeat : Bind repeated data HTML elements. Its like a for loop.
ng-if : Bind HTML elements with condition.

ng-show : Used to show the HTML elements.
ng-hide : Used to hide the HTML elements.
ng-class : Used to assign CSS class.
ng-src : Used to pass the URL image etc.

Event Listeners
ng-click : Click event to bind on HTML elements.
ng-dbl-click
Mouse event listeners
ng-mousedown
ng-mouseup
ng-mouseenter
ng-mouseleave
ng-mousemove
ng-mouseover
Keyboard event listeners
ng-keydown
ng-keyup
ng-keypress
ng-change

48. What is Angular Expression?

Angular expressions are JavaScript-like code snippets that are usually placed in bindings such as {{ expression }}. For example, these are valid expressions in Angular:

{{ 3+4 }}
{{ a+b }}
{{ user.name }}
{{ items[index] }}

49. How Angular expressions are different from JavaScript expressions?

Angular expressions are like JavaScript expressions but there are few differences.

Context : In Angular, the expressions are evaluated against a scope object, while the JavaScript expressions are evaluated against the global window object.

Forgiving: In Angular expression evaluation is forgiving to null and undefined, while in JavaScript undefined properties generates TypeError or ReferenceError.

No Control Flow Statements: Loops, conditionals or exceptions cannot be used in an Angular expression.

No Comma And Void Operators: You cannot use , (comma) or void in an Angular expression. And You cannot create regular expressions in an Angular expression.

50. What is compilation process in Angular?

Once you have the markup, the AngularJS needs to attach the functionality. This process is called "compilation" in Angular. Compiling includes rendering of markup, replacing directives, attaching events to directives and creating a scope. The AngularJS has compiler service which traverses the

DOM looking for attributes. The compilation process happens in two phases.
Compilation : traverse the DOM and collect all of the directives and creation of the linking function.
Linking: combine the directives with a scope and produce a live view. The linking function allows for the attaching of events and handling of scope. Any changes in the scope model are reflected in the view, and any user interactions with the view are reflected in the scope model.
When you create a new directive, you can write compile and/or linking functions for it to attach your custom behavior.
To understand the compilation process of Angular, must read "The nitty-gritty of compile and link functions inside AngularJS directives".

51. What is scope in AngularJS?

A scope is an object that ties a view (a DOM element) to the controller. In the MVC framework, scope object is your model. In other words, it is just a JavaScript object, which is used for communication between controller and view.

52. What is $rootscope in AngularJS?

The $rootScope is the top-most scope. An app can have only one $rootScope which will be shared among all the components of an app. Hence it acts like a global variable. All other $scopes are children of the $rootScope. Since $rootScope is a global, which means that anything you add here, automatically becomes available in $scope in all controller. To add something in $rootScope, you need to use app.run function which ensures that it will run prior to the rest of the app. You may say that "run" function is like "main" method of angular app.

```
app.run(function($rootScope) {
$rootScope.name = "AngularJS";
});
```

And then you can use in your view. (via expression)

```
<body ng-app="myApp">
<h1>Hello {{ name }}!</h1>
</body>
```

53. What are controllers in AngularJS?

In Angular, a Controller is a JavaScript constructor function. 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. The job of a controller is to pass data from the model, to the view or the view can asks for something from the controller, and the controller turns to the model and takes that thing, and hands it back to the view.

```
var myApp = angular.module('myApp', []);
myApp.controller('MyController', ['$scope', function($scope) {
$scope.website = 'jquerybyexample.net';
}
]);
```

And then in your HTML using ng-controller directive,

```
<div ng-controller="MyController">
```

```
<h1>My website address is {{ website }}!</h1>
 </div>
```

54. What is the difference between $scope and scope in AngularJS?

$scope is used while defining a controller (see previous question) where scope is used while creating a link function for custom directive. The common part is that they both refers to scope object in AngularJS, but the difference is that $scope uses dependency injection where scope doesn't. When the arguments are passed-in via dependency injection, their position doesn't matter. So for example, a controller defined as ($scope as first parameter)

```
myApp.controller('MyController', ['$scope', function($scope, $http) {
```

OR ($scope is second parameter)

```
myApp.controller('MyController', ['$scope', function($http, $scope) {
```

In both the case, the postion of $scope doesn't matter to AngularJS. Because AngularJS uses the argument name to get something out of the dependency-injection container and later use it.

But in case of link function, the position of scope does matter because it doesn't uses DI. The very first parameter has to be scope and that's what AngularJS expects.

```
app.directive("myDirective", function() {
return {
scope: {};
 link: function(scope, element, attrs) {
 // code goes here.
}
 };
});
```

However you can name it anything of your choice. In below code, foo is your scope object.

```
 link: function(foo, bar, baz) {
// code goes here.
}
```

55. What is MVC Architecture in AngularJS?

In AngularJS, scope objects are treated as Model. The View is display of model that is your data. And the model gets initialized within a JavaScript constructor function, called Controller in AngularJS. Let take a look at below code to understand it better.

```
<!DOCTYPE html>
<html>
<head>
<script data-require="angular.js@*" data-semver="1.3.6"
src="https://code.angularjs.org/1.3.6/angular.js"></script>
 <link rel="stylesheet" href="style.css" />
<script>
var myApp = angular.module('myApp', []);
 myApp.controller('MyController', ['$scope',
 function($scope) {
```

```
$scope.website = 'jquerybyexample.net';
 }
]);
</script>
 </head>

<body ng-app="myApp">
 <div ng-controller="MyController">
 <h1>My website address is {{ website }}</h1>;
</div>
</body>
</html>
```

Here MyController is a Controller and $scope (Model) is populated within Controller. And later on in div element $scope object data is displayed (View).

56. Can we have nested controllers in AngularJS?

YES. We can create nested controllers in AngularJS. Nested controller are defined in hierarchical manner while using in View. Take a look at below code. Here the hierarchy is "MainCtrl -> SubCtrl -> SubCtrl1".

```
<div ng-controller="MainCtrl">
 <p>{{message}} {{name}}!</p>

<div ng-controller="SubCtrl">
<p>Hello {{name}}!</p>

 <div ng-controller="SubCtrl2">
 <p>{{message}} {{name}}! Your username is  {{username}}.</p>
 </div>
 </div>
 </div>
```

57. In case of nested controllers, does the $scope object shared across all controllers?

YES. The $scope object is shared across all controllers and it happens due to scope inheritance. Since the ng-controller directive creates a new child scope, we get a hierarchy of scopes that inherit from each other. So if we define a property on a parent scope, the child scope can manipulate the property. And if the property is not present in child scope, then parent scope property's value is used. Lets consider, the previous question HTML where there are 3 controllers. And here is the AngularJS code to define these controllers.

```
var myApp = angular.module('myApp', []);
myApp.controller('MainCtrl', ['$scope',
function($scope) {
 $scope.message = 'Welcome';
$scope.name = 'Jon';
 }
```

```
]);
myApp.controller('SubCtrl', ['$scope',
function($scope) {
 $scope.name = 'Adams';
}
 ]);
 myApp.controller('SubCtrl2', ['$scope',
function($scope) {
$scope.name = 'Ema';
$scope.username = 'ema123';
}
 ]);
```

You will see that the controller "SubCtrl2" doesn't have "message" property define but it is used in HTML. So in this case, the immediate parent scope property will be used. But immediate parent scope which is "SubCtrl" in this case, also doesn't have "message" property. So it again goes one level up and finds the property is present so it uses parent scope value for "message" property and displays it.

58. What are different ways of bootstrapping AngularJS?

It is neat and superheroic JavaScript MVW (Model View Whatever) Framework which allows to extend HTML capabilities with Directives, expression, two way binding. In this post, we will see different ways of bootstrapping AngularJS. Bootstrapping in AngularJS is nothing but just the initialization of Angular app.

There are two ways of bootstrapping AngularJS. One is Automatic Initialization and other is Manually using Script.

When you add ng-app directive to the root of your application, typically on the <html> tag or <body> tag if you want angular to auto-bootstrap your application.

```
<html ng-app="myApp">
```

When angularJS finds ng-app directive, it loads the module associated with it and then compile the DOM.

Another way to bootstrapping is manually initializing using script. Manual initialization provides you more control on how and when to initialize angular App. It is useful when you want to perform any other operation before Angular wakes up and compile the page. Below is the script which allows to manually bootstrap angularJS.

```
<script>
angular.element(document).ready(function() {
angular.bootstrap(document, ['myApp']);
});
</script>
```

If you have noticed, it is using document.ready which is great as it will make sure that before bootstrapping your DOM is ready. You don't need to inclued jQuery library reference here, as angularJS has within it. So angular.bootstrap function bootstrap angularJS to document. The second parameter is the name of module. You need to take care of following things while using manual process.

Remember angular.bootstrap function will not create modules on the go. So you need to have your modules define, before you manually bootstrap. So below is the correct approach. First define the module and then bootstrap.

```
 <script>
 angular.module('myApp', [])
.controller('MyController', ['$scope', function ($scope) {
 $scope.message= 'Hello World';
 }]);

angular.element(document).ready(function() {
 angular.bootstrap(document, ['myApp']);
});
 </script>
```

You cannot add controllers, services, directives, etc after an application bootstraps.

59. What does SPA (Single page application) mean?

SPA is a concept where rather loading pages from the server by doing post backs we create a single shell page or master page and load the webpages inside that master page.

60. How can we create a custom directive in Angular?

Till now we have looked in to predefined Angular directives like “ng-controller”,”ng-model” and so on. But what if we want to create our own custom Angular directive and attach it with HTML elements as shown in the below code.

```
<div id=footercompany-copy-right></div>
```

To create a custom directive we need to use the “directive” function to register the directive with angular application. When we call the “register” method of “directive” we need to specify the function which will provide the logic for that directive.

For example in the below code we have created a copy right directive and it returns a copy right text.

Please note “app” is an angular application object which has been explained in the previous sections.

```
app.directive('companyCopyRight', function ()
{
return
{
 template: '@CopyRight questpond.com '
 };
});
```

The above custom directive can be later used in elements as shown in below code.

```
<div ng-controller="CustomerViewModel">
<div company-copy-right></div>
```

61. Explain data binding in AngularJS.

According to AngularJS.org, “Data-binding in Angular apps is the automatic synchronization of data between the model and view components. The way that Angular implements data-binding lets you treat the model as the single-source-of-truth in your application. The view is a projection of the model at all times. When the model changes, the view reflects the change, and vice versa.”

There are two ways of data binding:

1. Data mining in classical template systems
2. Data binding in angular templates

62. What are directives in AngularJS?

A core feature of AngularJS, directives are attributes that allow you to invent new HTML syntax, specific to your application. They are essentially functions that execute when the Angular compiler finds them in the DOM. Some of the most commonly used directives are ng-app,ng-controller and ng-repeat.

The different types of directives are:

- Element directives
- Attribute directives
- CSS class directives
- Comment directives

63. What is Angular Expression? How do you differentiate between Angular expressions and JavaScript expressions?

Angular expressions are code snippets that are usually placed in binding such as {{ expression }} similar to JavaScript.

The main differences between Angular expressions and JavaScript expressions are:

- **Context :** The expressions are evaluated against a scope object in Angular, while Javascript expressions are evaluated against the global window
- **Forgiving:** In Angular expression, the evaluation is forgiving to null and undefined whereas in JavaScript undefined properties generate TypeError or ReferenceError
- **No Control Flow Statements:** We cannot use loops, conditionals or exceptions in an Angular expression
- **Filters:** In Angular unlike JavaScript, we can use filters to format data before displaying it

64. What is the difference between link and compile in Angular.js?

- Compile function is used for template DOM Manipulation and to collect all the directives.
- Link function is used for registering DOM listeners as well as instance DOM manipulation and is executed once the template has been cloned.

65. What are the characteristics of ‘Scope’?

Scope is an object that refers to the application model. It is an execution context for expressions. Scopes are arranged in hierarchical structure which mimic the DOM structure of the application. Scopes can watch expressions and propagate events. The characteristics of Scope are:

- Scopes provide APIs ($watch) to observe model mutations.
- Scopes provide APIs ($apply) to propagate any model changes through the system into the view from outside of the “Angular realm” (controllers, services, Angular event handlers).
- Scopes can be nested to limit access to the properties of application components while providing access to shared model properties. Nested scopes are either “child scopes” or “isolate scopes”. A “child scope” (prototypically) inherits properties from its parent scope. An “isolate scope” does not.
- Scopes provide context against which expressions are evaluated. For example {{username}} expression is meaningless, unless it is evaluated against a specific scope which defines the username property.