1. What are directives?

Ans: Directives are markers on a DOM element (such as an attribute, element name, comment or CSS class) that tell AngularJS's HTML compiler ($compile) to attach a specified behaviour to that DOM element (e.g. via event listeners), or even to transform the DOM element and its children. And they are nothing but JavaScript functions that manipulate and add behaviours to HTML DOM elements.

2. What are Custom directives?

Ans: Custom directives are used in AngularJS to extend the functionality of HTML. It can modify or creates a new behaviour in HTML.

Custom Directives:

1) A new HTML element (<my-directive></my-directive>).
2) An attribute on an element (<input type="text" my-directive/>).
3) As a class (<input type="text" class="my-directive"/>).
4) As comment (<!-- directive:my-directive-->).

3. Role of ng-app directive in AngularJS?

Ans:

1) ng-app is the directive that can be placed anywhere in the application. And that's going to be the entry point of the application to AngularJS HTML compiler ($compile) like a "Main()" function in any compile time languages like C#, Java or C++ etc...
2) We cannot any directive in angular js without the declaration of ng-app.
3) When you add ng-app directive to the root of your application, typically on the <html> tag or <body> tag it auto-bootstrap's your application.

4. What is difference between Auto-Bootstrapping and Manual-Bootstrapping?

Ans:

Auto Bootstrap:

1) Angular will automatically bootstrap the application when it sees the ng-app directive. We no need put more efforts to use angular directives.
2) 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.

Manual Bootstrap:

1) Method of controling the bootstrap process programmatically or manually using angular.boostrap() method.
2) Manual Bootstrap will provide more control over when and where angular to be initialized. So that the programmer can avoid deadlock situations b/w various script references when using Automatic Bootstrapping.
3) Should not use the ng-app directive if you use angular.bootstrap() method.

Script helps you to manually bootstrap:

```
<script>

angular.element(document).ready(function () {

angular.bootstrap(document, ['myApp']);

});

</script>
```

5. Can we declare one or more ng-app directives in a HTML document ?

Ans:

1) Every angular application must have one ng-app directive in HTML document. If one or more ng-app appears in HTML doc then the first appearance will be used.
2) To run multiple applications in an HTML document you must manually bootstrap them using **angular.bootstrap()** because only one AngularJS application can be automatically bootstrapped per HTML document.

6. What are the responsibilities of **ng-model** directive?

Ans:

1) Binding the view into the model.
2) Providing validation behavior.
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,
5) ng-untouched, ng-empty, ng-not-empty) including animations.
6) Registering the control with its parent form.

7. What is difference between ng-show/ng-hide and ng-if directives ?

Ans: ng-show/ng-hide will always insert the DOM element, but will display/hide it based on the condition. Where ng-if removes and add its duplicated element based on conditions.

8. What is the purpose of ng-repeat directive?

Ans: ng-repeat directive is used to repeat HTML statements. Ng-repeat works the same as for each loop in C#, Java or PHP on a specific collection item like an array.

```
Ex: <ul ng-init="names=['A','B','C']">

<li ng-repeat="name in names">{{name}}</li>

</ul>
```

9. Steps to create CustomDirective in AngularJS?

Ans:

step1: Initialize the angular application.

step2: Define the directive name, Prefer using two or three letter prefix (except ng) while naming directives to avoid collision with future HTML releases.

Ex:

```
myApp.directive("myDirective", function () {
return {
restrict: 'E',
template: '<h1>My First directive</h1>'
};
});
```

10. Best Practice of Creating CustomDirective?

Ans:

1) Prefer using two or three letter prefix (except ng) while naming directives to avoid collision with future HTML releases
2) Prefer storing HTML template code in a separate file and assign the path to templateUrl variable.
3) Use controller function when there is a need to expose APIs to other directives.
4) Use “link” function if there is a need to modify the DOM.
5) Create IsolatedScope to reuse the component.

11. What is the role of Transclude property in Creating Custom directives?

Ans: Transclude defines whether to (transfer and include) one document or part of a document into another document. It allows to include original inner content of the directive’s HTML mark-up in the destination mark-up (which is defined in the template).

1) Transclusion property in AngularJS custom directive can hold value either of “true”, “false” or “element”.
2) By default trasclude property is set to "false" by browser.To make use of it we need to set it to "true".

12. What is the difference between template and templateUrl?

Ans:

template: specifies the HTML content that should be added to the HTML result of the directive.

templateUrl: loads the template for custom Directive from external page.

13. What is $rootscope and its uses?

Ans:

1) Every application has a single root scope. All other scopes are descendant scopes of the root scope.
2) Scopes provide separation between the model and the view, via a mechanism for watching the model for changes. The **$rootScope** instance gets created when a new application is bootstrapped. An app can have only one root scope which will be shared among all components of an application.
3) It acts like a global variable.
4) They also provide event emission/broadcast and subscription facility.