1. What is a service?

Ans: Angular JS has more than 30 built in services. Service is a constructor function where the injector invokes the default constructor when a service gets invoked.

2. List few well known built in services in AngularJS?

Ans: Angular js has about 30 built in services, here are the five mostly used services.

1) $scope
2) $rootscope
3) $location
4) $q
5) $http

3. Why services?

Ans: Services are injected using dependency injection mechanism of angular js. Services are the javascript logical functions which are responsible for performing specific task. This improves the code maintainability and testability.

4. What happens when a services is injected?

Ans: Services are singleton objects, it gets lazy instantiated and only once per angular application. Using $injector they get created when angular app components needs them.

5. In how many ways angular js services can be registered?

Ans: Angular services can be created or registered in five different ways:

1) service() method
2) factory() method
3) provider() method
4) value() method.
5) constant() method.

6. What is $http service?

Ans: $http transfers the data between server and the application and it makes request to the server and sends response.

7. What are the methods that are existed in $http Service?

Ans:

a. .get()
b. .put()

c. .post()
d. .delete()
e. .jsonp()
f. .head()
g. .patch()

8. What is the use of $q service?

Ans: $q service first fashion is JQuery's deferred implementation and the other will returns the promises. Every new instance of deferred is constructed by calling $q.defer().

The purpose of the deferred object is to expose the associated Promise instance as well as APIs that can be used for signalling the successful or unsuccessful completion, as well as the status of the task.

9. What are the methods that are mostly used in $q service?

Ans:

1) defer()
2) resolve(value) -- resolves the derived promise
3) reject(reason) -- rejects the derived promise
4) notify(value) -- provides updates on the status of the promise's execution

10. What is provider?

Ans:

1) The $provide service has a number of methods for registering components with the $injector. Many of these functions are also exposed on angular.Module.
2) The service providers are constructor functions. When instantiated they must contain a property called $get, which holds the service factory function.
3) Service provider returns the registered provider instance.

11. What is constant?

Ans: constants are used to pass values at config phase considering the fact that value cannot be used to be passed during config phase.

Ex: myApp.constant('mykey', "Hello World");

12. What does ngDialog do?

Ans: ngDialog makes modal dialogs and popups for Angular.js applications. You can customize it as you want and there are no other dependencies.

13. What are the list of accessible methods that are used to open dialog window?

Ans: open(Options) , openConfirm(Options)

14. What is the difference between ng-Dialog open() and openConfirm()?

Ans:

open()-- when we are using open user can only open the dialog window but cannot allow user to confirm the message.

openConfirm() --- when we are using openConfirm() user can open the dialog window and can confirm the message as well.

The function returns a promise that is either resolved or rejected depending on the way the dialog was closed. We can use closeThisDialog() to close the dialog window.

15. What are the options that we can use with open()?

Ans:

1) template {string}
2) controller {string}
3) scope {object}
4) className {string}
5) showClose {Boolean}
6) closeByEscape {Boolean}
7) closeByDocument {Boolean}
8) data {String}
9) close(id)
10) closeAll()

16. What are specilaized objects?

Ans: The objects which are specific to a particular Angular Framework API are known as specialized objects. In angularjs Injector is the one which creates specialized objects. These objects can be one of the controllers, directives, filters or any animations.

17. What is Single Responsibility Principle (SRP)?

Ans: The single responsibility principle teaches that every object should have a single responsibility.

Ex: Controller - Its main responsibilty is to wire up the scope to view. If its also responsible for making ajax calls then its violating the SRP rule.

18. What is Dependency Inversion Principle (DIP)?

Ans: The DIP states that objects should depend on abstractions, not concretions. In languages like c# objects will get dependent on Interface instead of class. This allows you to have loosely coupled code and has the added benefit of making unit testing much easier.