**Angular JS**

1. Two-way data binding

* Data binding in AngularJS is the synchronization between the model and the view.
* When data in the model changes, the view reflects the change, and when data in the view changes, the model is updated as well. This happens immediately and automatically, which makes sure that the model and the view is updated at all times.

1. Scope object

* When you make a controller in AngularJS, you pass the $scope object as an argument:
* Scope is a special JavaScript object which plays the role of joining controller with the views.
* Scope contains the model data. In controllers, model data is accessed via $scope object.

1. $Controllers

* AngularJS application mainly relies on controllers to control the flow of data in the application.
* A controller is defined using ng-controller directive.
* A controller is a JavaScript object containing attributes/properties and functions. Each controller accepts $scope as a parameter which refers to the application/module that controller is to control.

1. Services

* In AngularJS, a service is a function, or object, that is available for, and limited to, your AngularJS application.
* Services are JavaScript functions and are responsible to do a specific task only. This makes them an individual entity which is maintainable and testable. Controllers, filters can call them as on requirement basis. Services are normally injected using dependency injection mechanism of AngularJS.
* AngularJS has about 30 built-in services ($location, $http, $timeout, $interval, $window, $route)

1. Factory

Using factory method, we first define a factory and then assign method to it. (Check services.html)

1. $compile

The compile function deals with transforming the template DOM. Since most directives do not do template transformation, it is not used often. The compile function takes the following arguments:

* tElement - template element - The element where the directive has been declared. It is safe to do template transformation on the element and child elements only.
* tAttrs - template attributes - Normalized list of attributes declared on this element shared between all directive compile functions.
* transclude - A transclude linking function: function (scope, cloneLinkingFn)

1. $http

The $http service is a core AngularJS service that facilitates communication with the remote HTTP servers via the browser's [XMLHttpRequest](https://developer.mozilla.org/en/xmlhttprequest) object or via [JSON](http://en.wikipedia.org/wiki/JSONP)

1. $q
2. $window

Window is globally available in JavaScript; it causes testability problems. In AngularJS we always refer to it through the $window service, so it may be overridden, removed or mocked for testing.

1. Nglf

The ngIf directive removes or recreates a portion of the DOM tree based on an {expression}. If the expression assigned to ngIf evaluates to a false value then the element is removed from the DOM, otherwise a clone of the element is reinserted into the DOM.

1. ngHide

The ngHide directive shows or hides the given HTML element based on the expression provided to the ngHide attribute.

1. ngShow

The ngShow directive shows or hides the given HTML element based on the expression provided to the ngShow attribute.

1. ngForm

Nestable alias of [form](https://docs.angularjs.org/api/ng/directive/form) directive. HTML does not allow nesting of form elements. It is useful to nest forms, for example if the validity of a sub-group of controls needs to be determined.

1. ngClick

ngClick directive allows you to specify custom behavior when an element is clicked.

1. ngClass

The ngClass directive allows you to dynamically set CSS classes on an HTML element by databinding an expression that represents all classes to be added.

1. ngBind

* The **ng-bind** directive binds the **inner HTML** of the <p> element to the application variable **name**.

1. ngApp

* The ng-app directive defines the root element of an AngularJS application.
* The ng-app directive will auto-bootstrap (automatically initialize) the application when a web page is loaded.

1. ngInclude

Fetches, compiles and includes an external HTML fragment.

1. ngModel

The ng-model directive binds the value of HTML controls (input, select, text area) to application data. The ng-model directive can also:

* Provide type validation for application data (number, email, required).
* Provide status for application data (invalid, dirty, touched, error).
* Provide CSS classes for HTML elements.
* Bind HTML elements to HTML forms.

1. ngPattern

ngPattern adds the pattern [validator](https://docs.angularjs.org/api/ng/type/ngModel.NgModelController#$validators) to [ngModel](https://docs.angularjs.org/api/ng/directive/ngModel). It is most often used for text-based [input](https://docs.angularjs.org/api/ng/directive/input) controls, but can also be applied to custom text-based controls.

1. ngSwitch

The ngSwitch directive is used to conditionally swap DOM structure on your template based on a scope expression. Elements within ngSwitch but

without ngSwitchWhen or ngSwitchDefault directives will be preserved at the location as specified in the template.

1. ngTransclude

Directive that marks the insertion point for the transcluded DOM of the nearest parent directive that uses transclusion.

1. $cacheFactory

Factory that constructs [Cache](https://docs.angularjs.org/api/ng/type/$cacheFactory.Cache) objects and gives access to them.

1. Filter

Filters are used to change modify the data and can be clubbed in expression or directives using pipe character.

* currency Format a number to a currency format.
* date Format a date to a specified format.
* filter Select a subset of items from an array.
* json Format an object to a JSON string.
* limitTo Limits an array/string, into a specified number of elements/characters.
* lowercase Format a string to lower case.
* number Format a number to a string.
* orderBy Orders an array by an expression.
* uppercase Format a string to upper case.

1. Router

A Router is responsible for mapping URLs to components.

* Routers and "Routing Component" instances have a 1:1 correspondence.
* The Router holds reference to one or more of Outlets.
* There are two kinds of Router: [RootRouter](https://docs.angularjs.org/api/ngComponentRouter/type/RootRouter) and [ChildRouter](https://docs.angularjs.org/api/ngComponentRouter/type/ChildRouter).

1. RouteParams

A map of parameters for a given route, passed as part of the [Component Instruction](https://docs.angularjs.org/api/ngComponentRouter/type/ComponentInstruction) to the Lifecycle Hooks, such as $routerOnActivate and $routerOnDeactivate.