Interview questions and answer of Jquery,Node and Angular js

JQUERY:

3. Which command will give a version of jQuery?
Ans-The command $.ui.version returns jQuery UI version.
4. What is the difference between find and children methods?
Ans-Find method is used to find all levels down the DOM tree but children find single level down the DOM tree
5. What is the difference between childrens and children methods?
Ans-Find method is used to find all levels down the DOM tree but children find single level down the DOM tree
6. What is jQuery connect?
Ans-A ' jQuery connect' is a plugin used to connect or bind a function with another function. Connect is used to execute function from any other function or plugin is executed.
7. How to use connect?
Ans-Connect can be used by downloading jQuery connect file from jQuery.com and then include that file in the HTML file. Use $.connect function to connect a function to another function.
8. What are the basic selectors in jQuery?
Ans-Following are the basic selectors in jQuery:

- Element ID
- CSS Name
- Tag Name
- DOM hierarchy

9. What is the difference between size and length of jQuery?
Ans-Size and length both returns the number of element in an object. But length is faster than the size because length is a property and size is a method.
10. Can we add more than one 'document.ready' function in a page?
Ans-Yes, we can add more than one document.ready function in a page. But, body.onload can be added once in a page.
11. Whether our own specific characters are used in place of $ in jQuery?
Yes, We can use our own variable in place of $ by using the method called no Conflict () metho
12. What are the four parameters used for jQuery Ajax method?
The four parameters are

- URL – Need to specify the URL to send the request
- type – Specifies type of request(Get or Post)
- data – Specifies data to be sent to server
- Cache – Whether the browser should cache the requested page
- async - true : false

Setting async to false means that the statement you are calling has to complete before the next statement in your function can be called.If you set async: true then that statement will begin it's

execution and the next statement will be called regardless of whether the async statement has completed yet.

ANGULAR JS

1.Explain AngularJS Application Life-Cycle?

Ans- Understanding the life cycle of an AngularJS application makes it easier to learn about the way to design and implement the code. Apps life cycle consists of following three phases- bootstrap, compilation, and runtime.These three phases of the life cycle occur each time a web page of an AngularJS application gets loaded in the browser. Let's learn about each of the three phases in detail:

- The Bootstrap Phase – In this phase, the browser downloads the AngularJS javascript library. After this, AngularJS initializes its necessary components and the modules to which the ng-app directive points. Now that the module has loaded, required dependencies are injected into it and become available to the code within that module.
- The Compilation Phase – The second phase of the AngularJS life cycle is the HTML compilation stage. Initially, when a web page loads in the browser, a static form of the DOM gets loaded. During the compilation phase, this static DOM gets replaced with a dynamic DOM which represents the app view. There are two main steps – first, is traversing the static DOM and collecting all the directives. These directives are now linked to the appropriate JavaScript functionality which lies either in the AngularJS built-in library or custom directive code.The combination of directives and the scope, produce the dynamic or live view.
- The Runtime Data Binding Phase – This is the final phase of the AngularJS application. It remains until the user reloads or navigates to a different web page. At this point, any changes in the scope get reflected in the view, and any changes in the view are directly updated in the scope, making the scope the single source of data for the view.

2.What Are Different Ways To Create Service In AngularJS?

Ans-There are 5 different ways to create services in AngularJS.

- Value
- Factory
- Service
- Provider
- Constant

3.What Is The Difference Between The $Watch, $Digest, And $Apply?

Answer.

In AngularJS $scope object is having different functions like $watch(), $digest() and $apply() and we will call these functions as central functions. The AngularJS central functions $watch(),

$digest(), and $apply() are used to bind data to variables in view and observe changes happening in variables.

$Watch()

The use of this function is to observe changes in a variable on the $scope. It triggers a function call when the value of that variable changes. It accepts three parameters: expression, listener, and equality object. Here, listener and equality objects are optional parameters.

$Digest()

This function iterates through all the watch list items in the $scope object, and its child objects (if it has any). When $digest() iterates over the watches, it checks if the value of the expression has changed or not. If the value has changed, AngularJS calls the listener with the new value and the old value.

The $digest() function is called whenever AngularJS thinks it is necessary. For example, after a button click, or after an AJAX call. You may have some cases where AngularJS does not call the $digest() function for you. In that case, you have to call it yourself

$Apply()

AngularJS automatically updates the model changes which are inside AngularJS context. When you apply changes to any model, that lies outside of the Angular context (like browser DOM events, setTimeout, XHR or third-party libraries), then you need to inform the Angular about the changes by calling $apply() manually. When the $apply() function call finishes, AngularJS calls $digest() internally, to update all data bindings.

Following are the key differences between $apply() and $digest().

Its use is to update the model properties forcibly.

The $digest() method evaluates the watchers for the current scope. However, the $apply() method is used to evaluate watchers for root scope, that means it's for all scopes.

4.How would you specify that a scope variable should have one-time binding only?

Ans-By using "::" in front of it. This allows you to check if the candidate is aware of the available variable bindings in AngularJS

5.Explain what is a $scope and life cycle in AngularJS?

Ans-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. Scopes are objects that refer to the model. They act as glue between controller and view.This question is important as it will judge a developer's knowledge about a $scope object, and it is one of the most

important concepts in AngularJS. Scope acts like a bridge between view and model.

Scope Life Cycle

- Creation
- Watcher registration
- Model mutation
- Mutation observation
- Scope destruction

6.What are the differences between service and factory methods?

Factory Provider

Gives us the function's return value ie. You just create an object, add properties to it, then return that same object.When you pass this service into your controller, those properties on the object will now be available in that controller through your factory. (Hypothetical Scenario)

- Singleton and will only be created once
- Reusable components
- Factory are a great way for communicating between controllers like sharing data.
- Can use other dependencies
- Usually used when the service instance requires complex creation logic
- Cannot be injected in `.config()` function.
- Used for non configurable services
- If you're using an object, you could use the factory provider.
- Syntax: `module.factory('factoryName', function)`

Service Provider

- Gives us the instance of a function (object)- You just instantiated with the ‘new’ keyword and you’ll add properties to ‘this’ and the service will return ‘this’.When you pass the service into your controller, those properties on ‘this’ will now be available on that controller through your service. (Hypothetical Scenario)
- Singleton and will only be created once
- Reusable components
- Services are used for communication between controllers to share data
- You can add properties and functions to a service object by using the `this` keyword
- Dependencies are injected as constructor arguments
- Used for simple creation logic

- Cannot be injected in `.config()` function.
- If you're using a class you could use the service provider
- Syntax: `module.service('serviceName', function);`

7.What Is "$RootScope" In AngularJS?

Every application has a single root scope. All other scopes are descendant scopes of the root scope. Scopes provide separation between the model and the view, via a mechanism for watching the model for changes. They also provide event emission/broadcast and subscription facility.

8.What Is $RouteProvider In AngularJS?

$routeProvider is the key service which set the configuration of URLs, map them with the corresponding HTML page or ng-template, and attach a controller with the same.

9.What Is Data Binding? How Many Types Of Data Binding Directives Are Provided By AngularJS?

Ans- Data binding is the connection bridge between view and business logic (view model) of the application. Data binding in AngularJs is the automatic synchronization between the model and view. When the model changes, the view is automatically updated and vice versa. AngularJs support one-way binding as well as two-way binding.
AngularJS provides the following data binding directives:

- <ng-bind>
- <ng-bind-html>
- <ng-bind-template>
- <ng-non-bindable>
- <ng-model>

10.What Is Singleton Pattern? How Does Angular Use It?

A singleton pattern is an approach that we adopt to limit the instantiation of a Class to have only one object. In Angular, the dependency injection and the services implement the singleton pattern.Technically, if we call the **"new Class()"** two times without following the singleton pattern, the outcome will be two objects of the same class. Whereas a singleton enabled class will create the object first time and return the same object onwards.