css selectors link : http://www.nareshit.com/lesson/types-of-selectors/

Good Angularjs videos : http://codedamn.com/videos/angular/1/

**http://www.dwmkerr.com/promises-in-angularjs-the-definitive-guide/**

**Agile software** development is a group of **software** development methods in which solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change.

* Daily stand-ups,
* Planned meetings,
* Demo,
* Review,
* Retrospective Meetings, and
* To facilitate team meetings and decision-making process.

**Q) Angular compile process.**

Angular uses $compiler service to compile your Angular HTML page. The Angular compilation process begins after your HTML page (static DOM) is fully loaded.

**Component in Angulerjs 1.3+**

In AngularJS, a Component is a special kind of [directive](https://docs.angularjs.org/guide/directive) that uses a simpler configuration which is suitable for a component-based application structure.

**Advantages of Components:**

simpler configuration than plain directives

optimized for component-based architecture

writing component directives will make it easier to upgrade to Angular

**Single Page Application**(SPA) is a web **application** that fits on a **single** web **page** with dynamic actions without refreshing the **page**. ... **Single Page Application** can improve performance in several ways like loading time, using AJAX, easy to navigate **pages** etc.

## Advantages :

**Single Page Application** is good for making [Responsive Websites](http://www.zymphonies.com/blog/make-your-website-responsive), Support mobile, Tablet & Desktop.

* No extra queries to the server to download pages.
* User friendly.
* Performance Improvement, **Single Page Application** can improve performance in many ways, Single time file load each of HTML, CSS, JS.

## Disadvantages:

* Client must enable JavaScript, Single Page Application build with JavaScript, So JavaScript should be enabled in client browser. JavaScript enabled in all modern browsers by default.
* **Security**: Compare to traditional page Single Page Application is less secure due to Cross-site scripting (XSS).
* **Memory Leak**: Memory leak in JavaScript can even cause powerful system to slow down.

**http://www.javatpoint.com/javascript-global-variable**

**1.what is angularJs?**

AngularJs is a javascript framework that helps build web application. google company is the developed the angularjs.

**2.Advantages of AngularJs?**

Dependency Injection is a software design pattern in which components are given their dependencies instead of hard coding them within the component.

AngularJS comes with a built-in dependency injection mechanism. You can divide your application into multiple different types of components which AngularJS can inject into each other.

AngularJS provides a supreme Dependency Injection mechanism. It provides following core components which can be injected into each other as dependencies.

* value
* factory
* service
* provider
* constant
* **Two way binding** : - Keep the model and view in sync at all times, that is a change in the model update view and a change in the view update the model.
* Testing
* Client side MVC framework.
* Direvtives, Filters, routes

**One-Way Data Binding :**

In **One-Way** data binding, view (UI part) not updates automatically when data model changed and we need to write custom code to make it updated every time. Its not a synchronization processes

ng-bind has one-way data binding ($scope --> view).

Eg., ng-bind="myText" OR {{ myText }}

**Two-Way Data Binding :**

In **Two-way** data binding, view (UI part) updates automatically when data model changed. Its synchronization processes

ng-model is intended to be put inside of form elements and has two-way data binding ($scope --> view and view --> $scope)

e.g. <input name="firstname" ng-model="firstname"/>

## Disadvantages of AngularJS : -

* **Not Secure** − Being JavaScript only framework, application written in AngularJS are not safe. Server side authentication and authorization is must to keep an application secure.
* **Not degradable** − If your application user disables JavaScript then user will just see the basic page and nothing more.

**Core Features : --** Data-binding , Scope, Controller, Services, Filters, Directives, Templates.

**What is data binding in AngularJS?**

Data binding is the automatic synchronization of data between model and view components. ng-model directive is used in data binding.

**Q3. When First AngularJS was released?**  
2009  
**Q4. What is latest version of AngularJS?**  
v1.5.8

**Q5.Who created AngularJS?**  
Misko Hevery started to work on AngularJS in 2009. He was employee of Google.  
**Q6.Is it opensource?**  
Yes, It is free to use.  
**Q7. Explain what are the key features of Angular.js?**

1. Scope
2. Controller
3. Model
4. View
5. 7Services
6. Data Binding
7. Directives
8. Filters
9. Testable

**8. What is Module and how to write the module?**

Module is a container for different parts of your application ie. controllers,services,directives,filters etc...

var mainApp = angular.module("mainApp", []);

The module is a container for the application controllers.

Controllers always belong to a module.

**9.why are using module?**

Module as a main() method in other types of applications.

**10. What is controller?**

Controller is javascript function . the job of the controller is to build a model for the view to dispaly.

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.

**11. What is $scope and $rootScope?**

The scope is the binding part between the HTML (view) and the JavaScript (controller).

The scope is an object with the available properties and methods.

The scope is available for both the view and the controller.

Scope contains the model data. In controllers, model data is accessed via $scope object. $rootScope is the parent of all of the scope variables.

## Isolate scope:

**Isolated scopes** are the scopes which are fully dedicated to a particular custom directive only. It means that the scope inside a directive gets separated from the scope outside.

If you want to make a reusable directive you can’t rely on the parent scope and must use something called **Isolate Scope**instead.

inside scope "&" // read only one way binding example : brand : "&"

@ : Text bind

The @ character works well for accessing a string value passed into a directive.

inside scope "=" // read two way binding

You can use “=” character if you need to create a two-way binding between the outer **scope** and the directive's **isolate scope**.

http://www.journaldev.com/7572/angularjs-isolate-scope-two-way-binding-example

**12.What is directive?**

AngularJS lets you extend HTML with new attributes called **Directives**.

AngularJs have some default directives ng-app, ng-init,ng-model etc...

Custom **directives** are used in **AngularJS** to extend the functionality of HTML.

## What is a Service?

In AngularJS, a service is a function, or object, that is available for, and limited to, your AngularJS application.

**AngularJS** - Services. Services are normally injected using dependency injection mechanism of **AngularJS**.

AngularJS has about 30 built-in services. One of them is the $location service.

The $location service has methods which return information about the location of the current web page:

AngularJS provides many inbuilt services for example, $https:, $route, $window, $location etc. Each service is responsible for a specific task for example, $https: is used to make ajax call to get the server data. $route is used to define the routing information and so on.

**$http methods:**

* .delete():Removes all the current representations of the target resource given by URI.

delete(url, [config]);

* .get()
* .head():Same as GET, but it transfers the status line and the header section only.
* .jsonp()
* .patch()
* .post()
* .put():Replaces all the current representations of the target resource with the uploaded content.

## Properties

The response from the server is an object with these properties:

* .config the object used to generate the request.
* .data a string, or an object, carrying the response from the server.
* .headers a function to use to get header information.
* .status a number defining the HTTP status.
* .statusText a string defining the HTTP status.

**why use services?**

**Persist and share data between Controllers.**

**13. What are the services in AngularJS?**

AngularJS come with several built-in services. For example $http service is used to make XMLHttpRequests (Ajax calls). Services are singleton objects which are instantiated only once in app.

Services :

$timeout Service

$interval Service

$location

**Rules to enable the validation.**

#1. we should use either ng-form or form.

#2. form must be having a name attribute and declare novalidate.

--control level--

#3. control must be having name property.

#4. control must be having ng-model

#5. vailidator.

ng-required , ng-pattern, ng-minlength, ng-maxlength

**014. What are the filters in AngularJS?**

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

Filters select a subset of items from an array and return a new array. Filters are used to show filtered items from a list of items based on defined criteria.

Filters can be used with a binding expression or directive.

To apply the filter use pipe | character.

filters can do 3 different things

1.sort : accending ,decending

2.format : uppercase, lowercase,date,currency

3.filter data : limit to filter :-can be used to limit the number of rows and characters.

**syntax: --limitTo : number rows : start with row**

**15.What is MVC?**

**M**odel **V**iew **C**ontroller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

* **Model** − It is the lowest level of the pattern responsible for maintaining data.
* **View** − It is responsible for displaying data to the user.
* **Controller** − controls the interactions between the Model and View.

**16. Explain ng-init directive.**

The ngInit directive allows you to evaluate an expression in the current scope.

or

ng-init directive initializes an AngularJS Application data. It is used to put values to the variables to be used in the application.

17. **Explain what is injector?**  
An injector is a service locator, used to retrieve object instances.

**18. Explain what are factory method in angularJs?**  
Factory method are used to create the directive. It is invoked only once, when compiler matches the directive for the first time.

**Q. Which are the core directives of AngularJS?**

Following are the three core directives of AngularJS.

* ng-app
* ng-model
* ng-bind

**19.What is ng-app, ng-init and ng-model?**  
**ng-app** - It is the most important directive for an Angular Application, which is used to indicate starting of an Angular Application.

**ng-init** - To initialize the Angular Application data. or The ngInit directive allows you to evaluate an expression in the current scope.

**ng-model** - To bind the html tags (input, select, textarea) to Angular Application Data.

**ng-repeat** : ng-repeat directive repeats html elements for each item in a collection.

Because the ng-repeat directive repeats a block of HTML code for each item in an array, it can be used to create options in a dropdown list, but the ng-options directive was made especially for filling a dropdown list with options, and has at least one important advantage:

Dropdowns made with ng-options allows the selected value to be an **object**, while dropdowns made from ng-repeathas to be a string.

**ng-bind:** ng-bind directive binds the AngularJS Application data to HTML tags. ng-bind updates the model created by ng-model directive to be displayed in the html tag whenever user input something in the control or updates the html control's data when model data is updated by controller.

**ng-controller** : ng-controller directive tells AngularJS what controller to use with this view. 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.

**ng-disabled** : ng-disabled directive disables a given control.

**ng-show:** ng-show directive shows a given control.

**ng-hide:** ng-hide directive hides a given control.

**ng-click :** ng-click directive represents a AngularJS click event.

**20. How to avoid duplicates on ng-repeat**

Ans : track by

**21.What is routing in AngularJS?**

If you want to navigate to different pages in your application, but you also want the application to be a SPA (Single Page Application), with no page reloading, you can use the ngRoute module.

It is concept of switching views. AngularJS based controller decides which view to render based on the business logic.

**22. Explain 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.

Templates are the rendered view with information from the controller and model. These can be a single file (like index.html) or multiple views in one page using "partials".

eg : template: '<div ng-include="contentUrl"></div>'

**23. How to make an ajax call using Angular JS?**

AngularJS provides $http control which works as a service to make ajax call to read data from the server. The server makes a database call to get the desired records. AngularJS needs data in JSON format. Once the data is ready, $http can be used to get the data from server.

**24. What is use of $routeProvider in AngularJS?**

$routeProvider is the key service which set the configuration of urls, maps them with the corresponding html page or ng-template, and attaches a controller with the same.

**25.What is provider?**

provider is used by AngularJS internally to create services, factory etc. during config phase(phase during which AngularJS bootstraps itself).

The provider function of the $provide service takes two parameters: the name of the service and the function. A provider function must have a $get function. Return an object literal from the $get function

var module = angular.module("myModule",[]);

module.provider("myProvider", function(){

this.$get = function(){

return "welcome to india".

}

})

**26. Is AngularJS extensible?**

Yes! In AngularJS we can create custom directive to extend AngularJS existing functionalities.

**27. What is internationalization?**

Internationalization is a way to show local specific information on a website. For example, display content of a website in English language in United States and in Danish in France.

**28.difference between routeprovider and stateprovider in angularjs**

**$routeProvider :-**

$routeProvider is the key service which set the configuration of urls, maps them with the corresponding html page or ng-template, and attaches a controller with the same.

• The Page can contain only one ng-view on Page.

• $routeProvider don’t know about Parent/Child relationship.

It’s simple and easier to use.

**$stateProvider:-**

It works similar to $routeProvider but it purely focus on state rather than URLs.

• You can have multiple ui-view on single page

• Various view can be nested in each other and maintain by defining state in routing phase.

• We can have child & parent relationship here, simply like inheritance in state, also you could define sibling states.

**29. $watch, $watchGroup and $watchCollection angularjs**

• $watch method is for watching single scope properties.

A watch means that AngularJS watches changes in the variable on the $scope object.

• $watchCollection() method is for watching a scope property of collection in nature like Array.

• $watchGroup() method is for watching a group of scope properties in AngularJS application

$watch is used to watch expressions which can be either strings or functions (string expressions are parsed into functions). A string expression can not contain {{}}'s.

$scope.$watch("searchByModel", function (newVal, oldVal) {

console.log("Old Value is: " + oldVal);

console.log("New Value is: " + newVal)

});

$observe is a method on the attrs object and can only be used to observe the value change of a DOM attribute.

$q : A service that helps you run functions asynchronously, and use their return values (or exceptions) when they are done processing.

**what is promise?**

A promise is a special type of Object that we can either use, or construct ourselves to handle asynchronous tasks.

A promise has three states, pending, resolved or rejected.

In JavaScript, asynchronous methods usually use callbacks in order to inform a success or a failure state.

**$timeout**

The $timeout service can be used to call another JavaScript function after a given time delay. The $timeout service only schedules a single call to the function.

## $interval

The $interval service is similar in function to the $timeout service, except it schedules a function for repeated execution with a time interval in between.

**30. what is $digest and $apply?**

**$apply()**

$apply() is the function which is responsible to execute the entire list of watchers of all available scope in the application by invoking $digest on rootScope

or

The $scope.$apply() function takes a function as parameter which is executed, and after that $scope.$digest() is called internally. That makes it easier for you to make sure that all watches are checked, and thus all data bindings refreshed.

setTimeout(function () {

$scope.searchByModel="WagonR";

$scope.$apply();

},3000);

**What is a digest cycle in AngularJS?**

In each digest cycle Angular compares the old and the new version of the scope model values. The digest cycle is triggered automatically. We can also use $apply() if we want to trigger the digest cycle manually

if you need to run a $digest loop to update your DOM.

**digest link example :** https://www.youtube.com/watch?v=-H8YA3y4au0

$scope.$watch("searchByModel", function (newVal, oldVal) {

console.log("Old Value is: " + oldVal);

console.log("New Value is: " + newVal)

});

setTimeout(function () {

$scope.searchByModel="WagonR";

$scope.$apply();

},1000);

vehicles-controller.js

**What is $broadcast and $emit?**

That means $broadcast() sends an even downwards from parent to child controllers. The $emit() method, on the other hand, does exactly opposite. It sends an event upwards from the current controller to all of its parent controllers. From the syntax point of view a typical call to $broadcast() and $emit() will look like this:

$scope.$broadcast("MyEvent",data);

$scope.$emit("MyEvent",data);

**https://www.codementor.io/angularjs/tutorial/angularjs-interview-questions-sample-answers**

**31.How to create custom directives?**

Element directive : <custom-brand></custom-brand>

Attribute: <div custom-brand></div>

Class: <div class="custom-brand"></div>

comment : (M FOR comment)

or

1. Element directives(E)
2. Attribute directives(A)
3. CSS class directives(C)
4. Comment directives(M)

If you don't specify the restrict property it will default to "A"

restrict is for defining the directive type, and it can be A (Attribute), C (Class), E (Element), and M(comment) ,

An element directive is activated when AngularJS finds a matching HTML element in the HTML template. An attribute directive is activated when AngularJS finds a matching HTML element attribute. A CSS class directive is activated when AngularJS finds a matching CSS Class. And, a comment directive is activated when AngularJS finds a matching HTML comment.

(function () {

angular.module("components")

.directive("customHeader", [function () {

return {

templateUrl: "app/templates/header.html",

restrict: "A",

controller: 'headerCtrl',

link: function (scope, element, attrs) {

console.log(scope);

console.log(element);

console.log(attrs);

}

};

}]);

})();

**http://tutorials.jenkov.com/angularjs/custom-directives.html**

**32.what is Expressions?**

AngularJS expressions can be written inside double braces: {{ expression }}.

AngularJS expressions can also be written inside a directive: **ng-bind="expression".**

AngularJS will resolve the expression, and return the result exactly where the expression is written.

**33.** **How do you share data between controllers?**

Create an AngularJS service that will hold the data and inject it inside of the controllers.

However, there are couple of other ways to implement data sharing between controllers, like:

 Using events  
– Using $parent, nextSibling, controllerAs, etc. to directly access the controllers  
– Using the $rootScope to add the data

**Q). diff b/w ngRoute and ui-router?**

1. ngRoute is developed by the Angular team where as ui-router is a 3rd party module

2. ng-route implements routing based on the URL where as ui-router implements routing based on the state of the application

**difference between ng if and ng show**

The ng-if directive removes the HTML element if the expression evaluates to false.

If the if statement evaluates to true, a copy of the Element is added in the DOM.

The ng-show directive shows the specified HTML element if the expression evaluates to true, otherwise the HTML element is hidden.

**how to increase performance in angularjs**

1. If possible, use binding once
2. Stop using repeated filters
3. Avoid Watchers as much as possible
4. If possible, avoid ng-repeat
5. Limit your use of DOM filters
6. Use $watchCollection instead of $watch

**How do you reset $timeout in AngularJS.**

var customTimeout = $timeout(function () {

// arbitrary code

}, 55);

$timeout.cancel(customTimeout);

**$interpolate :** This service is used to evaluate angular expressions. You can run an entire string against a scope, and interpolate will give the result. e.g would be

var string = 'My Name is {{name}}';

$scope.name = 'Manish';

$interpolate(string)($scope); //this will result in My Name is Manish

**$parse :** This service is used as a getter/setter for single variables only. e.g would be

$scope.text = 'abc';

$parse('text')($scope); //this will result in abc

$parse('text').assign($scope,'xyz');

**HTML5**

**http://www.careerride.com/HTML5-Interview-Questions.aspx**

**Q . why are you using doctype in html?**

it is an instruction to the web browser about what version of the markup language the page is written in. The **doctype** declaration refers to a **Document Type** Definition (DTD)

**What happens if I dont put a <!DOCTYPE html> in my code?**

For example: new features & tags in HTML5 such as < article >,< footer >, < header >, < nav >,

< section > may not be supported if the Doctype is not declared.

Additionally, the browser may decide to automatically go into Quirks or Strict Mode

**Q). Difference b/w span tag and div tag**

The **HTML** <**span**> tag is used for grouping and applying styles to inline elements. There is a difference between the **span** tag and the div tag. The **span** tag is used with inline elements whilst the div tag is used with block-level content.

## Difference between HTML div tag and span tag

|  |  |
| --- | --- |
| **div tag** | **span tag** |
| HTML div is a **block** element. | HTML span is an **inline** element |
| HTML div element is used to **wrap large sections of elements**. | HTML span element is used to **wrap**  **small portion of texts, image** etc. |

**How to create dynamic div?**

var para = document.createElement("div");

var t = document.createTextNode("This is a paragraph.");

para.appendChild(t);

**Examples of block-level elements:**

<address>,<article>,<aside>,<blockquote>,<br>,<canvas>,<dd>,<div>,<dl>,<fieldset>,<figcaption>,<figure>,<footer> ,<form>,<h1>, <h2>, <h3>, <h4>, <h5>,<h6>,<header>,<hgroup>,<hr>,<li>,<main>,<nav>,<noscript>,<ol>,<output>,<p>,<pre>,<section>,<table>,<tfoot>,<ul>,<video>

**Examples of inline elements:**

* [**<a>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a) **,**[**<b>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/b)**,** [**<big>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/big)**,** [**<i>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/i)**,** [**<small>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/small)**,** [**<tt>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/tt)**,** [**<abbr>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/abbr)**,** [**<acronym>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/acronym)**,** [**<cite>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/cite)**,** [**<code>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/code)**,** [**<dfn>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dfn)**,** [**<em>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/em)**,**
* [**<kbd>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/kbd)**,**[**<strong>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong)**,**[**<samp>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/samp)**,**[**<time>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/time)**,**[**<var>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/var)**,**[**<a>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a)**,** [**<bdo>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/bdo)**,** [**<br>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/br)**,** [**<img>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img)**,** [**<map>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/map)**,** [**<object>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/object)**,** [**<q>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/q)**,**
* [**<script>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script)**,** [**<span>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span)**,**[**<sub>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/sub)**,**[**<sup>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/sup)**,** [**<button>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button)**,**[**<input>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input)**,**[**<label>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/label)**,**[**<select>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/select)**,**[**<textarea>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/textarea)**.**

**Q. Difference Between “Block” and “Inline”**

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

An inline element does not start on a new line and only takes up as much width as necessary.

**Some of the new features provided in HTML5 are:**

1. It provides support for local storage

2. New form controls, like calendar, date, time, email, url, search

3. <canvas> element is provided to facilitate 2D drawing

4. The <video> and <audio> elements are provided for media playback

5. New content-specific elements are provided. For e.g. <article>, <header>, <footer>, <nav>, <section>

**diff b/w html and html5**

Audio and Video are not part of HTML4

Audio and Videos are integral part of HTML5 e.g. <audio> and <video> tags.

Vector Graphics is possible with the help of technologies such as VML, Silverlight, Flash etc

Vector graphics is integral part of HTML5 e.g. SVG and canvas

Not possible to draw shapes like circle, rectangle, triangle.

Using Html5 you can draw shapes like circle, rectangle, triangle.

**sizes** : desktop : 1024 \* 627

tab : 768 \* 627

mobile : 360 \*415

**How many ways to create layout?**

There are 3 ways to design layout of a web page:

1. **HTML Div with CSS**: fast and widely used now.
2. **HTML Table**: slow and less preferred.
3. **HTML Frameset**: deprecated now.

**The Bootstrap grid system has four classes:**

* xs (for phones)
* sm (for tablets)
* md (for desktops)
* lg (for larger desktops)

The classes above can be combined to create more dynamic and flexible layouts.

**Q) What is Carousel?**

The Carousel plugin is a component for cycling through elements, like a carousel (slideshow).

## Basic Pagination?

If you have a web site with lots of pages, you may wish to add some sort of pagination to each page.

To create a basic pagination, add the .pagination class to an <ul> element:

**Q. Core elements in html**

The four core attributes that can be used on the majority of HTML elements (although not all) are:

1. id.

2. title.

3. class.

4. style.

**Q. difference b/w display none and visibility hidden.**

**display: none** removes the element completely from the document. It does not take up any space,

**visibility: hidden** hides the element, but it still takes up space in the layout.

**1.What is canvas?**

The HTML5 canvas element can be used to draw graphics on the webpage via scripting

Canvas has several methods for drawing paths, boxes, circles, characters, and adding images.

Html5 canvas element is used to create : Graphs and Charts, Animations, Games, Diagrams, Videos and Photo galleries, Special image effects, Drawing applications, user interface elements.

**2.What is SVG?**

**S**calable **V**ector **G**raphics commonly known as SVG is a XML based format to draw vector images. It is used to draw two dimentional vector images.

* SVG stands for Scalable Vector Graphics
* SVG is used to define graphics for the Web

**3.Diff b/w canvas and svg ?**

| **SVG** | **Canvas** |
| --- | --- |
| Vector based (composed of shapes) or shape based | Raster based (composed of pixel) or pixel based |
| Multiple graphical elements, which become the part of the DOM | Single HTML element similar to [<img>](http://www.tutorialrepublic.com/html-tutorial/html5-svg.php) in behavior |
| Modified through script and CSS | Modified through script only |
| Give better performance with smaller number of objects or larger surface, or both | Give better performance with smaller surface or larger number of objects, or both |
| Better scalability — can be printed with high quality at any resolution | Poor scalability — not suitable for printing on higher resolution |

**4.What is local storage?**

With local storage, web applications can store data locally within the user's browser.

Before HTML5, application data had to be stored in cookies, included in every server request. Local storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

// Store  
localStorage.setItem("lastname", "Smith");

sessionStorage.setItem("lastname", "Smith");

or

// Store  
localStorage.lastname = "Smith";

sessionStorage.lastname = "Smith";

//get Item propery

localStorage.getItem(lastname );

sessionStorage.getItem(lastname );

//deleting property

localStorage.removeItem("lastname");

sessionStorage.removeItem("lastname");

//Clearing the Local Storage

sessionStorage.clear();

localStorage.clear();

**Reading Number of Properties Stored**

You can read the number of properties stored in the sessionStorage or localStorage objects using the length property, like this:

var length = sessionStorage.length;

var length = localStorage.length;

**The storageEvent event object passed to the event handler function looks like this:**

Attaching an event listener to a local storage object is done like this:

function onStorageEvent(storageEvent){

alert("storage event");

}

window.addEventListener('storage', onStorageEvent, false);

The function onStorageEvent() is the event handler function.

The addEventListener() function call attaches the event handler function to storage events.

The storageEvent event object passed to the event handler function looks like this:

StorageEvent {

key; // name of the property set, changed etc.

oldValue; // old value of property before change

newValue; // new value of property after change

url; // url of page that made the change

storageArea; // localStorage or sessionStorage,

// depending on where the change happened.

}

**Cookies:**

Good for small amount of data

Cookies are not recommended for long-term storage of information in a browser for these reasons.

if you want to support most browsers, then do not exceed 50 cookies per domain, and 4093 bytes per domain. That is, the size of all cookies should not exceed 4093 bytes. This means you can have 1 cookie of 4093 bytes, or 2 cookies of 2045 bytes, etc.

## 5. The sessionStorage Object?

The sessionStorage object is equal to the localStorage object, **except** that it stores the data for only one session. The data is deleted when the user closes the specific browser tab.

**6. What is Application Cache?**

HTML5 introduces application cache, which means that a web application is cached, and accessible without an internet connection.

Application cache gives an application three advantages:

1. Offline browsing - users can use the application when they're offline

2. Speed - cached resources load faster

3. Reduced server load - the browser will only download updated/changed resources from the server

**7. What is a Web Worker?**

When executing scripts in an HTML page, the page becomes unresponsive until the script is finished.

A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.

## 8. What does *HTML5 Cookie* mean?

An HTML 5 cookie is a cookie-like storage options available in HTML 5. It consists of browser-based local storage and session storage, which is created and accessible by the Web page itself.

**9.What is indexeddb?**

The indexeddb is a new HTML5 concept to store the data inside user's browser. indexeddb is more power than local storage and useful for applications that requires to store large amount of the data. These applications can run more efficiency and load faster.

**10.whta is web sql db?**

The Web SQL Database API isn't actually part of the HTML5 specification but it is a separate specification which introduces a set of APIs to manipulate client-side databases using SQL.

## What is a Manifest file?

A Manifest file is a simple text file that tells the browser what to cache and what not to cache.  
  
There are three sections of a Manifest file:  
  
1) **CACHE MANIFEST -** Files listed here are cached after they are downloaded for the first time.  
2) **NETWORK -** Files listed here require a connection to the server, and are never cached.  
3) **FALLBACK -** Files listed here specify fallback pages if a page is inaccessible.

**GULP**

**What is gulp?**

Gulp is a task runner that uses Node.js as a platform. It purely uses the JavaScript code and helps to run front-end tasks and large-scale web applications. Gulp builds system automated tasks like CSS and HTML minification, concatenating library files, and compiling the SASS files. These tasks can be run using Shell or Bash scripts on the command line.

**Why to use Gulp?**

It is shorter, simpler and faster as compared to other task runner.

Uses SASS and LESS as CSS preprocessor.

Automatically refreshes page after editing the source files.

Easy to understand and build the Gulpfile.js because, it uses pure JavaScript code to build the task.

* **first we need to install nodejs** :
* if you want minification the css and html gulp can be helped
* First we need remove old versions gulp if any case already install in system.

1.npm rm gulp -g

npm : node package manager

rm : remove

-g : remove any where on your computer

2.npm install gulp-cli -g

gulp-click : gulp command line interface

javaScript

**1. What is JavaScript?**

JavaScript is a client-side as well as server side scripting language that can be inserted into HTML pages and is understood by web browsers. JavaScript is also an Object Oriented Programming language

or

JavaScript is an object-based scripting language that is lightweight and cross-platform.

**2. Where JavaScript is used?**

JavaScript is used to create interactive websites. It is mainly used for:

Client-side validation

Dynamic drop-down menus

Displaying data and time

Displaying popup windows and dialog boxes (like alert dialog box, confirm dialog box and prompt dialog box)

Displaying clocks etc.

JavaScript provides 3 places to put the JavaScript code: within body tag, within head tag and external JavaScript file.

**What's the difference between putting script in head and body?**

When scripts are included in the head they load or run before the content of the page. When you include them in the body they load or run after the preceding html. It's usually good practice to put scripts as close to the end of the body as possible.

Placing scripts at the bottom of the <body> element improves the display speed.

scripts are generally placed at the END of <body> - simply due to the fact that the scripts run when loaded. If this is in <head> then the script will run early in the loading process and can cause delays in the page loading through.

**What are the advantages of using JavaScript?**

There are mainly two advantages of JavaScript functions.

1. **Code reusability**: We can call a function several times so it save coding.
2. **Less coding**: It makes our program compact. We don’t need to write many lines of code each time to perform a common task.
3. Form validation can be done on the client side, which reduces the unnecessary round trips b/w the client and the server

**What are disadvantages of using JavaScript?**

**JavaScript can not be used for Networking applications because there is no such support available.**

**JavaScript doesn't have any multithreading or multiprocess capabilities.**

**InnerText property is supported in IE and Chrome. But not in Firefox. we need to replace the innerText to textContext it will be work IE, Chrome and Firefox.**

**ref link : https://www.youtube.com/watch?v=iMsvZNhvpkg&index=3&list=PL6n9fhu94yhUA99nOsJkKXBqokT3MBK0b**

**String Methods :**

**isLowerCase**

**isUpperCase**

**indexOf**

**lastiIdexOf**

**cancat**

**Array Methods:**

**push() :**

**pop() :**Removes the last element from an array and returns that element.

**shift() :--** method removes the first item of an array, and returns that item.

**unshift()**

**delete arr[0];**

**join();**

**http://www.w3schools.com/jsref/jsref\_obj\_string.asp**

**Exceptions in Javascript**

The **try** statement lets you test a block of code for errors.

The **catch** statement lets you handle the error.

The **throw** statement lets you create custom errors.

The **finally** statement lets you execute code, after try and catch, regardless of the result.

**$exceptionHandler :-** AngularJS also includes built-in $exceptionHandler service, which handles uncaught exceptions in the application.

**Javascript performance?**

Reduce Activity in Loops

## Avoid Unnecessary Variables

## Reduce DOM Size

## Reduce DOM Access

**Opp's in javascript :**

Inheritance

polymorphisam

Abstraction

encapsulation

**Q). data types in Javascript**

1. Primitive data type
2. Non-primitive (reference) data type

## JavaScript primitive data types

|  |  |
| --- | --- |
| String | represents sequence of characters e.g. "hello" |
| Number | represents numeric values e.g. 100 |
| Boolean | represents boolean value either false or true |
| Undefined | represents undefined value |
| Null | represents null i.e. no value at all |

## JavaScript non-primitive data types

The non-primitive data types are as follows:

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| Object | represents instance through which we can access members |
| Array | represents group of similar values |
| RegExp | represents regular expression |

**Q). Creating objects in javascript?**

1. By object literal
2. By creating instance of Object directly (using new keyword)
3. By using an object constructor (using new keyword)

## 1) JavaScript Object by object literal

The syntax of creating object using object literal is given below:

1. object={property1:value1,property2:value2.....propertyN:valueN}

As you can see, property and value is separated by : (colon).

Let’s see the simple example of creating object in JavaScript.

1. **<script>**
2. emp={id:102,name:"Shyam Kumar",salary:40000}
3. document.write(emp.id+" "+emp.name+" "+emp.salary);
4. **</script>**

## 2. By creating instance of Object

The syntax of creating object directly is given below:

1. var objectname=new Object();

Here, **new keyword** is used to create object.

Let’s see the example of creating object directly.

1. **<script>**
2. var emp=new Object();
3. emp.id=101;
4. emp.name="Ravi Malik";
5. emp.salary=50000;
6. document.write(emp.id+" "+emp.name+" "+emp.salary);
7. **</script>**

## 3) By using an Object constructor

Here, you need to create function with arguments. Each argument value can be assigned in the current object by using this keyword.

The **this keyword** refers to the current object.

The example of creating object by object constructor is given below.

1. **<script>**
2. function emp(id,name,salary){
3. this.id=id;
4. this.name=name;
5. this.salary=salary;
6. }
7. e=new emp(103,"Vimal Jaiswal",30000);
9. document.write(e.id+" "+e.name+" "+e.salary);
10. **</script>**

**Adding and deleting values in javascript**

<!DOCTYPE html>

<html>

<body>

<p id="demo"></p>

<script>

var cars = ["Saab","Volvo","BMW"]

cars.push("abc")

document.getElementById("demo").innerHTML = cars[0];

delete cars[3];

document.getElementById("demo").innerHTML = cars[2];

</script>

</body>

</html>

**Q). How to check text field is empty or not?**

if( document.myForm.EMail.value == "" )

{

alert( "Please provide your Email!" );

}

var emailID = document.myForm.EMail.value;

atpos = emailID.indexOf("@");

dotpos = emailID.lastIndexOf(".");

### What is the difference between == and ===?

The == operator checks equality only whereas === checks equality and data type i.e. value must be of same type.

**What is the use of isNaN function?**

isNan function returns true if the argument is not a number otherwise it is false.

**Which company developed JavaScript?**

Netscape is the software company who developed JavaScript.

**What is a prompt box?**

A prompt box is a box which allows the user to enter input by providing a text box.  Label and box will be provided to enter the text or number.

**Explain the working of timers in JavaScript? Also elucidate the drawbacks of using the timer, if any?**

Timers are used to execute a piece of code at a set time or also to repeat the code in a given interval of time. This is done by using the functions **setTimeout, setInterval**and**clearInterval**.

The **setTimeout(function, delay)** function is used to start a timer that calls a particular function after the mentioned delay. The **setInterval(function, delay)** function is used to repeatedly execute the given function in the mentioned delay and only halts when cancelled. The **clearInterval(id)** function instructs the timer to stop.

**What is the difference between ViewState and SessionState?**

‘ViewState’ is specific to a page in a session.

‘SessionState’ is specific to user specific data that can be accessed across all pages in the web application.

**Explain how can you submit a form using JavaScript?**

To submit a form using JavaScript use document.form[0].submit();

document.form[0].submit();

**What are all the types of Pop up boxes available in JavaScript?**

* Alert
* Confirm and
* Prompt

diff b/e alert, confirm and prompt.?

http://www.programminginterviews.info/2011/06/differentiate-between-alert-prompt-and.html

**Explain how to read and write a file using JavaScript?**

There are two ways to read and write a file using JavaScript

* Using JavaScript extensions
* Using a web page and Active X objects

**How can you convert the string of any base to integer in JavaScript?**

The parseInt() function is used to convert numbers between different bases. parseInt() takes the string to be converted as its first parameter, and the second parameter is the base of the given string.

**How can you convert the number to string in JavaScript?**

var num = 15;  
var n = num.toString();

**What is called Variable typing in Javascript?**

Variable typing is used to assign a number to a variable and the same variable can be assigned to a string.

Example

i = 10;

i = "string";

**15.diff between attr and prop methods.**

Attributes are refering to additional information of an object.

Properties are describing the characteristics of an object.

In an object oriented context, Attribute refers to a class of objects and Property refers to an instant.

<button id="myBtn" onclick="myFunction()">Try it</button>

<p>The result should be 2 (the button element's id and onclick attributes).</p>

<p id="demo"></p>

<script>

function myFunction() {

var x = document.getElementById("myBtn").attributes.length;

document.getElementById("demo").innerHTML = x;

}

</script>

*objectName.property*// person.age

**16.what is keyframes who to use it.**

The @keyframes rule specifies the animation code. The animation is created by gradually changing from one set of CSS styles to another. During the animation, you can change the set of CSS styles many times. ... 0% is the beginning of the animation, 100% is when the animation is complete.

**17.diff between transforms and transition.**

CSS Transitions allows property changes in CSS values to occur smoothly over a specified duration.

ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)

linear - specifies a transition effect with the same speed from start to end

ease-in - specifies a transition effect with a slow start

ease-out - specifies a transition effect with a slow end

ease-in-out - specifies a transition effect with a slow start and end

**CSS transforms** allows elements styled with CSS to be transformed in two-dimensional or three-dimensional space.

* scaling—changing the size of the element
* rotating—spinning the element
* translating—moving the element

img { transform: rotate(30deg); }

**Q). what is protocal def and usage?**

The protocol property sets or returns the protocol of the current URL, including the colon (:).

The protocol is a standard that specifies how data are transmitted between computers.

# Q) JavaScript Object Prototypes?

Every JavaScript object has a prototype. The prototype is also an object.

All JavaScript objects inherit their properties and methods from their prototype.

using prototype we are inherit the classes.

https://www.youtube.com/watch?v=yXlFR81tDBM&index=62&list=PL6n9fhu94yhUA99nOsJkKXBqokT3MBK0b

how to use the prototype property to add a property to an object:

https://www.w3schools.com/jsref/jsref\_prototype\_math.asp

**Q) current version javascript**

1.8

## Q).JavaScript Scope

In JavaScript, objects and functions are also variables.

**In JavaScript, scope is the set of variables, objects, and functions you have access to.**

# Q).JavaScript Hoisting

## Hoisting is JavaScript's default behavior of moving all declarations to the top of the current scope

## Q). JavaScript Declarations are Hoisted

In JavaScript, a variable can be declared after it has been used.

or In JavaScript, a variable can be declared after assign the value or before assign the value.

* If a developer doesn't understand hoisting, programs may contain bugs (errors).
* To avoid bugs, always declare all variables at the beginning of every scope.

# JavaScript Use Strict

"use strict"; Defines that JavaScript code should be executed in "strict mode".

## The "use strict" Directive

The "use strict" directive is new in JavaScript 1.8.5 (ECMAScript version 5).

It is not a statement, but a literal expression, ignored by earlier versions of JavaScript.

The purpose of "use strict" is to indicate that the code should be executed in "strict mode".

**Q). diff b/w for loop and foreach ?**

The For Loop is best for iterating over name-value pairs, and the For Each Loop best for iterating over values, for example arrays.

**The For Loop**

var o = {'name':'Batman', 'age':33, 'city':'Gotham City'};  
for (var p in o) {  
    console.log(p+': '+o[p]);  
}

**The For Each Loop**

var animals = ['cat', 'dog', 'monkey', 'donkey'];  
for each (var v in animals) {  
    console.log(v);  
}

**1. What is BOM and DOM?**

The **Browser Object Model** (BOM) is used to interact with the browser.

The default object of browser is window means you can call all the functions of window by specifying window or directly.

The Document Object Model is a programming API for documents.

A Document object represents the HTML document that is displayed in that window.

The way a document content is accessed and modified is called the **Document Object Model.**

### What is the use of window object?

The window object is automatically created by the browser that represents a window of a browser.

It is used to display the popup dialog box such as alert dialog box, confirm dialog box, input dialog box etc.

**Q). Difference b/w undefined and null?**

**undefined** means a variable has been declared but has not yet been assigned a value. On the other hand, **null** is an assignment value. It can be assigned to a variable as a representation of no value.

null === undefined // false

null == undefined // true

null === null // true

**example : 1**

var TestVar;

alert(TestVar); //shows undefined

alert(typeof TestVar); //shows undefined

var TestVar = null;

alert(TestVar); //shows null

alert(typeof TestVar); //shows object

**example : 2**

**https://www.toptal.com/javascript/interview-questions**

(function(){

var a = b = 3;

})();

console.log("a defined? " + (typeof a !== 'undefined'));

console.log("b defined? " + (typeof b !== 'undefined'));

var b = 3;

var a = b;

But in fact, var a = b = 3; is actually shorthand for:

b = 3;

var a = b;

As a result (if you are not using strict mode), the output of the code snippet would be:

a defined? false

b defined? true

### Difference between Client side JavaScript and Server side JavaScript?

**Client side JavaScript** comprises the basic language and predefined objects which are relevant to running java script in a browser. The client side JavaScript is embedded directly by in the HTML pages. This script is interpreted by the browser at run time.

**Server side JavaScript** also resembles like client side java script. It has relevant java script which is to run in a server. The server side JavaScript are deployed only after compilation.

**2. what is callback function?**

A **JavaScript Callback Function** is a **function** that is passed as a **parameter** to another **JavaScript function**, and the **callback function** is run inside of the **function** it was passed into. or

JavaScript statements are executed line by line. However, with effects, the next line of code can be run even though the effect is not finished. This can create errors.

To prevent this, you can create a callback function.

A callback function is executed after the current effect is finished.

**3. what is closures?**

It is used to access outer function variables in inner function.

JavaScript variables can belong to the **local** or **global** scope.

**Q. difference between onchange and onclick in javascript**

***onchange*** — The event is triggered when a field has changed. This is especially useful for dropdown lists.

***onclick*** — The event is triggered when the field is clicked on. This is especially useful for radio buttons and checkboxes.

**Q. How typeof operator works?**

You can use the **typeof** operator to find the data type of a JavaScript variable. **or**

The typeof operator evaluates to "number", "string", or "boolean" if its operand is a number, string, or boolean value and returns true or false based on the evaluation.

**example:**

typeof(4+7); //returns number

typeof("4"+"7"); //returns string

typeof(4\*"7"); //returns number

typeof(4+"7"); //returns string

alert(typeof(myvar));

if (typeof xyz === "undefined")  
alert("xyz is undefined");

**4.what is public, private?**

* Private members should only be accessible within the scope.
* Public members should only be accessible outside the scope.

**5.what splice and slice?**

The **splice()**method returns the removed item(s) in an array and **slice()**method returns the selected element(s) in an array, as a new array object.

*array*.splice(index,howmany,item1, .....,itemX)

* **index** − An integer that specifies at what position to add/remove items, Use negative values to specify the position from the end of the array
* **howMany** − The number of items to be removed. If set to 0, no items will be removed
* **element1, ..., elementN** − The elements to add to the array. If you don't specify any elements, splice simply removes the elements from the array.

*array*.slice(start,end)

**6.diff b/w substr and substring?**

string.substr(beginIndex, length);

string.substring(beginIndex, endindex);

**7.what is JSON?**

JSON is a format for storing and transporting data.

JSON is often used when data is sent from a server to a web page.

* JSON stands for JavaScript Object Notation.
* JSON is lightweight data-interchange format.
* JSON is easy to read and write than XML.
* JSON is language independent.
* JSON supports array, object, string, number and values.

# JSON vs XML

A list of differences between JSON and XML are given below.

|  |  |  |
| --- | --- | --- |
| **No.** | **JSON** | **XML** |
| 1) | JSON stands for JavaScript Object Notation. | XML stands for eXtensible Markup Language. |
| 2) | JSON is simple to read and write. | XML is less simple than JSON. |
| 3) | JSON is easy to learn. | XML is less easy than JSON. |
| 4) | JSON is data-oriented. | XML is document-oriented. |
| 5) | JSON doesn't provide display capabilities. | XML provides the capability to display data because it is a markup language. |
| 6) | JSON supports array. | XML doesn't support array. |
| 7) | JSON is less secured than XML. | XML is more secured. |
| 8) | JSON files are more human readable than XML. | XML files are less human readable. |
| 9) | JSON supports only text and number data type. | XML support many data types such as text,  number, images, charts, graphs etc. Moreover,  XML offeres options for transferring the format  or structure of the data with actual data. |

***how to define json data.***

*File: first.json*

1. {"employees":[
2. {"name":"Sonoo", "email":"sonoojaiswal1987@gmail.com"},
3. {"name":"Rahul", "email":"rahul32@gmail.com"},
4. {"name":"John", "email":"john32bob@gmail.com"}
5. ]}

**8. what is Ajax?**

AJAX is an acronym for **Asynchronous JavaScript and XML**. It is a group of inter-related technologies like JavaScript, DOM, XML, HTML, CSS etc.

AJAX allows you to send and receive data asynchronously without reloading the web page. So it is fast.

### Where it is used in ajax?

There are too many web applications running on the web that are using ajax technology like **gmail**, **facebook**,**twitter**,**google map**, **youtube** etc.

**9.what is jquery?**

jQuery is a small and lightweight JavaScript Library.

jQuery greatly simplifies JavaScript programming.

jQuery is easy to learn

jQuery is cross-platform.

jQuery means "write less do more".

jQuery simplifies AJAX call and DOM manipulation.

### Why do we use jQuery?

* It is very easy to learn and use.
* It is used to develop browser compatible web applications.
* It improves the performance of an application.
* It is very fast and extensible.
* It facilitates you to write minimal lines of codes for UI related functions.
* It provides a cross-browser support.

**What are the basic selectors in jQuery?**

Following are the basic selectors in jQuery:

* Element ID
* CSS Name
* Tag Name
* DOM hierarchy

**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

## What is jQuery UI

* Good for highly interactive web applications
* Open source and free to use
* Powerful theme mechanism
* Stable and maintenance friendly
* Extensive browser support

## Do I need Jquery for Angular?

No , you do not need Jquery for Angular. It’s independent of Jquery.

**10.what is selector?**

jQuery selectors allow you to select and manipulate HTML element(s).

jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing [CSS Selectors](http://www.w3schools.com/cssref/css_selectors.asp), and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses: $().

# 11. jQuery Traversing Methods

first() : Returns the first element of the selected element

last() : Returns the last element of the selected element

next( ) : Returns the next sibling element of the selected element

nextAll : Returns the all next sibling element of the selected element

prev() : Returns the previous sibling element of the selected element

prevAll() : Returns the All Previous sibling element of the selected element

**refer link : http://www.w3schools.com/jquery/jquery\_ref\_traversing.asp**

**12. diff b/w padding and margin?**

1. Padding is considered internal to the element, margin is external to the element.

2. Padding adds to the width and height of the element. (unless you use box-sizing:border-box; to override this behaviour). Margin does not affect the dimensions of the element.

3. Margins can handle an extra unit value: auto. Padding can't use auto as a value.

4. margin:auto; is really useful for centering a block element within it's container. Padding can only center the element's contents.

**CSS**

**Diff between CSS and CSS3.**

**CSS**

CSS stands for Cascading Style Sheets

CSS describes how HTML elements are to be displayed on screen, paper, or in other media

CSS saves a lot of work. It can control the layout of multiple web pages all at once

External stylesheets are stored in CSS files

**CSS3**

CSS3 Adds Rounded Corners and Gradients

CSS3 Adds Animation Features and Text Effects

**Some of the most important CSS3 modules are:**

Selectors, Box Model, Backgrounds and Borders, Image Values and Replaced Content, Text Effects,

2D/3D Transformations, Animations, Multiple Column Layout, User Interface.

**CSS Selectors: why using selectors means : Code Reuseablity**

**http://www.nareshit.com/lesson/types-of-selectors/**

Styles Reuseablity.

1.Tag/Type Selector : h1

2.ID Selector : (#) must be unique. Most not be duplicate. If duplicate first one is effected when working with javascript. In CSS Id selector is duplicate it's reflect but when your calling with script but problem will come.

3.Class Selector : (.) it is possible duplicate.

4.Grouping Selectors : h1,h2,h3...etc

5.customized selectors : <shashi>hiiii</shashi>

**Bootstrap CSS**

To ensure proper rendering and touch zooming, add the viewport meta tag to your <head>.

<meta name="viewport" content="width=device-width, initial-scale=1">

You can disable zooming capabilities on mobile devices by adding user-scalable=no to the viewport meta tag. This disables zooming, meaning users are only able to scroll, and results in your site feeling a bit more like a native application. Overall, we don't recommend this on every site, so use caution!

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">

**http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-fixed-layout.php**

**CSS3 Layouts :**

**Bootstrap Fixed Layout :** The process of creating the fixed yet responsive layout starts with the .container class.

The columns can be created inside the rows using the predefined grid classes like .col-xs-\*, .col-sm-\*, .col-md-\* and .col-lg-\* where \* represent grid number and should be from 1 to 12.

Use **.container** for a responsive fixed width container.

<div class="container">

...

</div>

**Bootstrap Fluid Layout :**

In Bootstrap (version 3.2+), you can use the class .container-fluid to create the fluid layouts in order to utilize the 100% width of the viewport.

Use **.container-fluid** for a full width container, spanning the entire 100% width of your viewport.

<div class="container-fluid">

...

</div>

**What is Bootstrap Grid System**

Bootstrap grid system provides the quick and easy way to create responsive website layouts.

Bootstrap 3 includes predefined grid classes for quickly making grid layouts for different types of devices like cell phones, tablets, laptops and desktops, etc. For example, you can use the .col-xs-\* class to create grid columns for extra small devices like cell phones, similarly the .col-sm-\* class for small screen devices like tablets, the .col-md-\* class for medium size devices like desktops and the .col-lg-\* for large desktop screens.

<div class="row">

<div class="col-md-8">.col-md-8</div>

<div class="col-md-4">.col-md-4</div>

</div>

**Q. How to add scrollbar both sides in css3**

overflow-x: scroll;

overflow-y: scroll;

or

overflow:scroll

if you want add horizental scrollbar

overflow-y:hidden

if you want add vertical scrollbar

overflow-x:hidden

**Q. How to make image in responsive.**

img{

max-width : 100%;

height : auto;

}

**Media queries**

The @media query is used to define different style rules for different media types/devices.

**http://getbootstrap.com/css/**

We use the following media queries in our Less files to create the key breakpoints in our grid system.

/\* Extra small devices (phones, less than 768px) \*/

/\* No media query since this is the default in Bootstrap \*/

/\* Small devices (tablets, 768px and up) \*/

@media (min-width: @screen-sm-min) { ... }

/\* Medium devices (desktops, 992px and up) \*/

@media (min-width: @screen-md-min) { ... }

/\* Large devices (large desktops, 1200px and up) \*/

@media (min-width: @screen-lg-min) { ... }

We occasionally expand on these media queries to include a max-width to limit CSS to a narrower set of devices.

@media (max-width: @screen-xs-max) { ... }

@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }

@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }

@media (min-width: @screen-lg-min) { ... }

### What is the CSS Box model and what are its elements?

The CSS box model is used to define the design and layout of elements of CSS.

The elements are:

* Margin
* Border
* Padding
* Content

**what is z-index?**

The z-index property specifies the stack order of an element.

### How can you integrate CSS on a web page?

There are three methods to integrate CSS on web pages.

1. Inline method
2. Embedded/Internal method
3. Linked/Imported/External method

**diff b/w id and class?**

### ID's are unique

* Each element can have only one ID
* Each page can have only one element with that ID

### Classes are NOT unique

* You can use the same class on multiple elements.
* You can use multiple classes on the same element.

Deliotte

**how to insert html5 into angularjs**

**how find the angularjs application slow**

Q. form elements html5

<form> , <input>, <textarea> , <label>, <fieldset>, <legend>, <select>, <optgroup>,<option>, <button>,<datalist>,<keygen>,<output>

Q. input types in html5

type text, number, submit, button, reset, checkbox, radio

**how to change location $location service**

**diff b/w services and factory**

factories are functions that return the object, while services are constructor functions of the object which are instantiated with the new keyword.

1.If you're using an object, you could use the factory

If you're using a class you could use the service

2.Factories are used to hold reusable code.

Services are used to access external services.

So if you need to create a text formatting module that will be used site wide or an authentication module, a factory is a good place for it.

If you need an authorization module or to access a data store or server put it in a service.

Syntax: **module**.factory('factoryName', **function**);

Syntax: **module**.service(‘serviceName’, **function**);

**Common features**

* Singleton and will only be created once
* Reusable component
* Used for communication between controllers to share data
* Cannot be injected in .config() function.

**provider example :**

var module = angular.module("myModule",[]);

module.provider("myProvider", function(){

this.$get = function(){

return "welcome to india".

}

})

**factory example:**

var module = angular.module("myModule",[]);

module.factory("myProvider", function(){

return "welcome to india".

})

mod.controller("MyController", function(myProvider) {

console.log("MyController - myProvider: " + myProvider);

});

**service exampel :**

var module = angular.module("myModule",[]);

module.service("myService", function() {

this.getValue = function() {

return "My Value";

};

});

**directive example :**

(function () {

angular.module("components")

.directive("customHeader", [function () {

return {

templateUrl: "app/templates/header.html",

restrict: "A",

controller: 'headerCtrl',

link: function (scope, element, attrs) {

console.log(scope);

console.log(element);

console.log(attrs);

}

};

}]);

})();

**1.diff b/w src and ngsrc?**

when ever we r using binding expression that time compulsary we need to use ng-src.

if you r not using ng-src instead using src we r getting error 404 Error.

**2.how to display alert box in center?**

<!DOCTYPE html>

<html>

<style>

#content {

position: absolute;

top: 50%;

left: 50%;

width: 400px;

height: 200px;

margin-top: -100px;

margin-left: -200px;

background-color: #DDD;

}

</style>

<body>

<div id="content">

Some content

</div>

</body>

</html>

**3.how to know ajax file is loaded or not**

**4.what is resolve in route concept?**

**5.how to insert value in array particular index position?(Splice)**

**6.In which scenerio we r using $apply?**

**7.what is isolated scope and scenerio?**

**8.how to stop time interval after 5 seconds.**

http://jsfiddle.net/eMBW2/

//Html Code

<div id="txt">

</div>

//Javascript Code

var counter = 0;

var tt=setInterval(function(){startTime()},1000);

function startTime()

{

if(counter == 10) {

clearInterval(tt);

} else {

counter++;

}

document.getElementById('txt').innerHTML = counter;

}

**9.diff b/w normal page and single page?**

**10.without scope how to get values in html page?**

**11.what is ng-init?**

The ngInit directive allows you to evaluate an expression in the current scope.

**12.what is broad casting and emitting?**

**13.what is web worker?**

**14.what html5 API?**

**angular 2.0 link tutorial**

**https://www.youtube.com/watch?v=-zW1zHqsdyc**

**difference b/w Angualrjs 1.0 and Angular2.0**

No more controllers and scope

components / reuseable code

reduced learning curve

type script & ES2015 / E16

better mobile support

**what is type script?**

A strict superset of javascript with added features.

Maintained by microsoft

Optional static typing

class based object oriented programming

Resembles language like java and c/c++.