1. How to add and remove plugin in ionic?

Add-

$ ionic cordova plugin [<action>] [<plugin>]

$ ionic cordova plugin add cordova-plugin-inappbrowser@latest

Remove-

$ ionic cordova plugin rm cordova-plugin-camera

2. How to run Ionic project on browser?

$ ionic serve

3. How to create model in bootstrap?

<div class="container">

<h2>Modal Example</h2>

<!-- Trigger the modal with a button -->

<button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#myModal">Open Modal</button>

<!-- Modal -->

<div class="modal fade" id="myModal" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Modal Header</h4>

</div>

<div class="modal-body">

<p>Some text in the modal.</p>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

4. Can we use multiple time header and footer semantic element in one html?

Yes We can use header footer tag multiple in code but its depends upon to add the header and footer in one div or another div its depends upon coding.

5. Ranking of style get html?

1.Using !important

div{

background-color:yellow !important;

color:red !important;

padding:100px !important;

}

2.Inline css -

style="background-color:black;color:white;padding:20px;"

3. ID css

#b{

background-color:white;color:black;padding:60px;

}

4. Class css

.a{

background-color:red;color:green;padding:50px;

}

5. Element css.

div{

background-color:green;color:red;padding:70px;

}

6. how can we use em unit in css?

Another way of setting the font size is with em values. The size em value is dynamic. When defining the font-size property, an em is equal to the size of the font that applies to the parent of the element in question. If you haven't set the font size anywhere on the page, then it is the browser default, which is often 16px. So, by default 1em = 16px, and 2em = 32px. If you set a font-size of 20px on the body element, then 1em = 20px and 2em = 40px. Note that the value 2 is essentially a multiplier of the current em size.

In order to calculate the em equivalent for any pixel value required, you can use this formula:

em = desired element pixel value / parent element font-size in pixels

7. what is the use of $apply in angular js.

For example we have create one two way data binding program in angular, if we have and ng-model in input and getting data to controller. Default all function have watcher , and we can use $watch default service also in watcher there is two value old value and new value, so all function work asynchronous so if backend provide some value to $scope in that function but function is not recognise the value because function execute asynchronous if we want this value the we have to use $apply service there and getting all watcher value in that function.

8.$Bradcast and $emit work?

$broadcast :- Data share Parent controller to child controller

$emit :-Data share child to parent controller

9. $apply and $digest difference?

$apply() and $digest() have some similarities and differences. They are similar in that they both check what's changed and update the UI and fire any watchers.

One difference between the two is how they are called. $digest() gets called without any arguments. $apply() takes a function that it will execute before doing any updates.

The other difference is what they affect. $digest() will update the current scope and any child scopes. $apply() will update every scope. So most of the time $digest() will be what you want and more efficient.

The final difference which explains why $apply() takes a function is how they handle exceptions in watchers. $apply() will pass the exceptions to $exceptionHandler (uses try-catch block internally), while $digest() will require you handle the exceptions yourself.

10. What is local storage and session storage in Html 5?

**localStorage**: stores data with no expiration date, and gets cleared only through JavaScript, or clearing the Browser Cache / Locally Stored Data

**sessionStorage**: similar to localStorage but expires when the browser closed (not the tab).

**Cookie**: stores data that has to be sent back to the server with subsequent requests. Its expiration varies based on the type and the expiration duration can be set from either server-side or client-side (normally from server-side).

Cookies are primarily for **server-side** reading (can also be read on client-side), localStorage and sessionStorage can only be read on **client-side.**

**HTML web storage provides two objects for storing data on the client:**

* **window.localStorage - stores data with no expiration date**
* **window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)**

**11. Json convert to string?**

**Stringfy ---->**

**JSON.stringify(this.posts);**

**12. What is JSON.parse?**

**Parse the data with JSON.parse(), and the data becomes a JavaScript object.**

**13.2d animation in css**

**div {**

**-ms-transform: translate(50px, 100px); /\* IE 9 \*/**

**-webkit-transform: translate(50px, 100px); /\* Safari \*/**

**transform: translate(50px, 100px);**

**}**

**14. 3d animation in css**

**#myDiv {**

**-webkit-transform: rotateX(150deg); /\* Safari \*/**

**transform: rotateX(150deg);**

**}**

**15. What is the bootstrap in angularjs?**

**16. Inline and block level elememt**

**Block level elements in HTML:**

[**<address>**](https://www.w3schools.com/tags/tag_address.asp) [**<article>**](https://www.w3schools.com/tags/tag_article.asp)

[**<aside>**](https://www.w3schools.com/tags/tag_aside.asp) [**<blockquote>**](https://www.w3schools.com/tags/tag_blockquote.asp) [**<canvas>**](https://www.w3schools.com/tags/tag_canvas.asp) [**<dd>**](https://www.w3schools.com/tags/tag_dd.asp) [**<div>**](https://www.w3schools.com/tags/tag_div.asp) [**<dl>**](https://www.w3schools.com/tags/tag_dl.asp) [**<dt>**](https://www.w3schools.com/tags/tag_dt.asp) [**<fieldset>**](https://www.w3schools.com/tags/tag_fieldset.asp) [**<figcaption>**](https://www.w3schools.com/tags/tag_figcaption.asp)[**<figure>**](https://www.w3schools.com/tags/tag_figure.asp) [**<footer>**](https://www.w3schools.com/tags/tag_footer.asp) [**<form>**](https://www.w3schools.com/tags/tag_form.asp) [**<h1>-<h6>**](https://www.w3schools.com/tags/tag_hn.asp) [**<header>**](https://www.w3schools.com/tags/tag_header.asp) [**<hr>**](https://www.w3schools.com/tags/tag_hr.asp) [**<li>**](https://www.w3schools.com/tags/tag_li.asp) [**<main>**](https://www.w3schools.com/tags/tag_main.asp) [**<nav>**](https://www.w3schools.com/tags/tag_nav.asp) [**<noscript>**](https://www.w3schools.com/tags/tag_noscript.asp) [**<ol>**](https://www.w3schools.com/tags/tag_ol.asp) [**<output**](https://www.w3schools.com/tags/tag_output.asp)**[>](https://www.w3schools.com/tags/tag_output.asp)**

[**<p>**](https://www.w3schools.com/tags/tag_p.asp)[**<pre>**](https://www.w3schools.com/tags/tag_pre.asp)[**<section>**](https://www.w3schools.com/tags/tag_section.asp)[**<table>**](https://www.w3schools.com/tags/tag_table.asp)[**<tfoot>**](https://www.w3schools.com/tags/tag_tfoot.asp)[**<ul>**](https://www.w3schools.com/tags/tag_ul.asp)[**<video>**](https://www.w3schools.com/tags/tag_video.asp)

**Inline elements in HTML:**

[**<a>**](https://www.w3schools.com/tags/tag_a.asp) [**<abbr>**](https://www.w3schools.com/tags/tag_abbr.asp) [**<acronym>**](https://www.w3schools.com/tags/tag_acronym.asp) [**<b>**](https://www.w3schools.com/tags/tag_b.asp)  [**<bdo>**](https://www.w3schools.com/tags/tag_bdo.asp) [**<big>**](https://www.w3schools.com/tags/tag_big.asp) [**<br>**](https://www.w3schools.com/tags/tag_br.asp) [**<button>**](https://www.w3schools.com/tags/tag_button.asp) [**<cite>**](https://www.w3schools.com/tags/tag_cite.asp) [**<code>**](https://www.w3schools.com/tags/tag_code.asp) [**<dfn>**](https://www.w3schools.com/tags/tag_dfn.asp) [**<em>**](https://www.w3schools.com/tags/tag_em.asp) [**<i>**](https://www.w3schools.com/tags/tag_i.asp) [**<img>**](https://www.w3schools.com/tags/tag_img.asp) [**<input>**](https://www.w3schools.com/tags/tag_input.asp)  [**<kbd>**](https://www.w3schools.com/tags/tag_kbd.asp)  [**<label>**](https://www.w3schools.com/tags/tag_label.asp)  [**<map>**](https://www.w3schools.com/tags/tag_map.asp)  [**<object>**](https://www.w3schools.com/tags/tag_object.asp)  [**<q>**](https://www.w3schools.com/tags/tag_q.asp)

[**<samp>**](https://www.w3schools.com/tags/tag_samp.asp)  [**<script>**](https://www.w3schools.com/tags/tag_script.asp)  [**<select>**](https://www.w3schools.com/tags/tag_select.asp)  [**<small>**](https://www.w3schools.com/tags/tag_small.asp)  [**<span>**](https://www.w3schools.com/tags/tag_span.asp)  [**<strong>**](https://www.w3schools.com/tags/tag_strong.asp)  [**<sub>**](https://www.w3schools.com/tags/tag_sub.asp)  [**<sup>**](https://www.w3schools.com/tags/tag_sup.asp)  [**<textarea>**](https://www.w3schools.com/tags/tag_textarea.asp) <[**time>**](https://www.w3schools.com/tags/tag_time.asp) [**<tt>**](https://www.w3schools.com/tags/tag_tt.asp)  [**<var>**](https://www.w3schools.com/tags/tag_var.asp)

**17. $http service in angular js.**

**get-method**

**$http({**

**method : "GET",**

**url : "welcome.htm"**

**}).then(function mySuccess(response) {**

**$scope.myWelcome = response.data;**

**}, function myError(response) {**

**$scope.myWelcome = response.statusText;**

**});**

**Post- method**

**var req = {  
 method: 'POST',  
 url: 'http://example.com',  
 headers: {  
 'Content-Type': undefined  
 },  
 data: { test: 'test' }  
}  
  
$http(req).then(function(){...}, function(){...});**

**18. Http service use in angular 2?**

**Get Method -**

getPosts(){

this.http.get(this.url)

.subscribe(res =>{

this.posts = res.json();

console.log(this.posts);

});

}

**post-method**

createPost(input:HTMLInputElement){

let post={title:input.value};

this.http.post(this.url,JSON.stringify(post)).

subscribe(res =>{

post['id'] = res.json().id;

this.posts.splice(0,0,post);

console.log(res);

});}

**19. Ionic 2 Details.**

**20. What is difference main difference between observable and promises.**

**21. Div come in center of the screen.**

**22. Str length found without using length.**

**23.promise cancel mean?**

1. **Alert Sumation alert(1+1+"5"); ===>25**

**alert("5"+5+6); ===> 556**

**console.log(1+1+"5") ===>25**

**console.log("5"+1+1) ===> 511**

**first in console will be addition next string it will be work as string**

1. **without change varilable set value**

**login to print 0 1 1 2 3 5 8 13**

1. **Difference betwwen AOT and JIT**
2. **Choosen.jQuery use drop down part?**

**11.Datatype two types premative**

1. **hosting in javascript**

**13.closer mean in javascript**

1. **prototype in javascript**
2. **inheritance in javascript**
3. **prime number in javascript**
4. **wordpress folder system**
5. **whats is mean by Angular js in details**
6. **whats is the directive why we are use directive in Angular.**
7. **what is services why we are use service in angular**

**21.bootstrap--> 22. create it using JS**

**\* \* \* \*  
 \* \*  
 \* \* \* \* \*  
 \* \*  
 \* \* \* \***

1. **what is mean by object in javascript**
2. **Angualr js structure.**

**----------------------------------------next page----------------------------------------**

**28/5/2018**

**Q1 What is closure in Javascript?**

Ans🡪 A closure is a combination of data and function in one unit(one function),

Closues are useful because they let you associate some data (the lexiacal environment) with a function that operates on that data.

function init() {

  var name = 'Mozilla'; // name is a local variable created by init

  function displayName() { // displayName() is the inner function, a closure

    alert(name); // use variable declared in the parent function

  }

  displayName();

}

init();

function makeFunc() {

  var name = 'Mozilla 1';

  function displayName() {

    alert(name);

  }

  return displayName;

}

var myFunc = makeFunc();

myFunc();

//A closure is the combination of a function and the lexical environment within which that function was declared.

// This environment consists of any local variables that were in-scope at the time the closure was created.

//In this case, myFunc is a reference to the instance of the function displayName created when makeFunc is run.

//The instance of displayName maintains a reference to its lexical environment, within which the variable name exists.

//For this reason, when myFunc is invoked, the variable name remains available for use and "Mozilla" is passed to alert.

function makeAdder(x) {

  return function(y) {

    return x + y;

  };

}

var add5 = makeAdder(5);

var add10 = makeAdder(10);

console.log(add5(2));  // 7

console.log(add10(2)); // 12

**Arrow function in TypeScript**

let noParamFn = () =>{console.log('No parameter function')}

let oneParamFn = (x) =>{console.log('One parameter function',x)}

let oneParamFnAlt1 = x=>console.log('One parameter function',x);

let oneParamFnAlt2 = (x:string) =>console.log('One explicit parameter function',x);

noParamFn();

oneParamFn('Arun');

oneParamFn(10);

oneParamFn({x:20,y:30});

oneParamFnAlt1(22);

oneParamFnAlt2('Lets play');

*A solid understanding of functional/block scope, anonymous functions, closures, and*[*IIFE’s*](https://stackoverflow.com/questions/8228281/what-is-the-function-construct-in-javascript)*will definitely make you a better JavaScript developer and help you out in future interviews.*

**Datatype in typeScript**

//here the tye is number i.e 1,2,3,3.0,3.44 etc

let a:number;

a=20;

//here type is string, it will accept any string value.

let b: string;

b='Arun';

//here type is array of number.

let c:number[]=[12,23,33,5];

//here type is an object

let d:{x:10,y:20};

//here type is any, by default type is any.it can accept

//anything like var keyword in javascript.

let e;

e=[12,3,4,5];

//here type is constant

  const f = 10;

//type is enum. to declare multiple constants

  enum color {red,green,blue};

  //to access enum

 console.log(color.blue);

**What is bootstrap in Angular ?**

**Ans :- In main.ts we have to include the bootstrap module**

import { enableProdMode } from '@angular/core';

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';

import { environment } from './environments/environment';

if (environment.production) {

  enableProdMode();

}

platformBrowserDynamic().bootstrapModule(AppModule)

  .catch(err => console.log(err));

**Its will assign in app.module.ts**

bootstrap: [AppComponent]

**In app.component.ts file selector “app-root” use in index.html**

**To initialize the angular app “boot – start - app”**

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>AngualrFirstBatch</title>

  <base href="/">

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

  <link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

  <app-root></app-root>

</body>

</html>

* *declarations*—this application's lone component.
* *imports*—import [BrowserModule](https://angular.io/api/platform-browser/BrowserModule) to have browser specific services such as DOM rendering, sanitization, and location.
* *providers*—the service providers. 🡪 services custom
* *bootstrap*—the *root* component that Angular creates and inserts into the index.html host web page.

1. Unit Testing test cases for developer in angular 2?

Ans :- tslinit.ts file will add the logic of coding. We can add rule for testing in this file.

In Package.Json file we are including installed extension in app with version

**Property Binding 🡪 [ ]**

**Binding in component to View**

import { Component, OnInit } from '@angular/core';

@Component({

  selector: 'app-prop-binding',

  template: `<div>

  <h1>{{title}}</h1>

  <img [src]="imageUrl">

</div>`,

  styleUrls: ['./prop-binding.component.css']

})

export class PropBindingComponent implements OnInit {

  title = "Angular Property binding examples";

  imageUrl = "./../assets/images/picture.jpg";

  constructor() { }

  ngOnInit() {

  }

}

**Attribute Binding:-**

  <td [attr.colSpan]="colSpan">Arun</td>

**Class binding: -**

<div>

  <span class="glyphicon "

  [class.glyphicon-star]="!isSelected"

  [class.glyphicon-star-empty]="isSelected"

  (click)="onClick()">

  </span>

</div>

**What is differnce between class and interface**

|  |  |
| --- | --- |
| **InterFace** | **Class** |
| interface Point {      x:number,      y:number,    }  let drawDistance = (point :Point) =>{        console.log(point.x);      console.log(point.y);  }  drawDistance({x:10,y:20}); | class Point{      x:number;      y:number;      //constructor is used to initialize the data member      constructor(x:number,y:number){          this.x = x;          this.y = y;      }      draw(){          console.log(this.x);          console.log(this.y);      }  }  let po = new Point(20,10);  po.draw(); |
| **Can’t add constructor** | **Can add constructor** |
| **Store the more than access  specifier** | **Can add interface in class** |
|  |  |

I use class for example in any class I want to add get method which return processed data, and when there is no behavior added to object and I want to directly access object I will use interface... Using class if you define constructor you restrict user to define certain variable to initialized before constructing any object.

**What is the difference between directive and component?**