1.How many types of **Data Binding**

<https://medium.com/@venkateshece1105/different-types-of-data-binding-in-angular-8-0-f84c17ca7eb0#:~:text=Types%3A%201%20String%20Interpolation%202%20Property%20Binding%2C,3%20Event%20Binding%2C%204%20Two%20Way%20Data%20Binding>.

<https://angular.io/guide/binding-syntax>

2.Parent to child how you will pass the data

[Sharing Data between Angular Components - Four Methods (fireship.io)](https://fireship.io/lessons/sharing-data-between-angular-components-four-methods/#:~:text=Sharing%20Data%20between%20Angular%20Components%20-%20Four%20Methods,Unrelated%20Components%3A%20Sharing%20Data%20with%20a%20Service.%20)

3.Difference between local storage and session storage

[Difference Between Local Storage, Session Storage And Cookies - GeeksforGeeks](https://www.geeksforgeeks.org/difference-between-local-storage-session-storage-and-cookies/#:~:text=Difference%20Between%20Local%20Storage%2C%20Session%20Storage%20And%20Cookies,server%20is%20exist%20%201%20more%20rows%20)

4.Difference between observable and promises

[Angular Promises Versus Observables | Syncfusion Blogs](https://www.syncfusion.com/blogs/post/angular-promises-versus-observables.aspx)

5.Reactive Form

[How To Use Reactive Forms in Angular | DigitalOcean](https://www.digitalocean.com/community/tutorials/angular-reactive-forms-introduction)

6.Get response from data using url in front end

[Angular Tutorial: Get JSON Data from API and Display in HTML – HowToCreateApps](https://howtocreateapps.com/angular-tutorial-json/)

7.Test cases - Karma.js

8. Difference between div && span

[tags - What is the difference between HTML div and span elements? - Stack Overflow](https://stackoverflow.com/questions/183532/what-is-the-difference-between-html-div-and-span-elements)

9.Difference between Position and relative

10. How angular application angular lifecycle hooks works

[Component Life Cycle Hooks in Angular - TekTutorialsHub](https://www.tektutorialshub.com/angular/angular-component-life-cycle-hooks/#:~:text=The%20life%20cycle%20hooks%20are%20the%20methods%20that,starts%20it%20creates%20and%20renders%20the%20root%20component.)

11. Types of directives

[Fundamentals of Angular Directives - Three Types of Directives in Angular - Thinkster](https://thinkster.io/tutorials/fundamentals-of-angular-directives-learn-the-three-types-of-directives)  
[Angular 8 - Directives (tutorialspoint.com)](https://www.tutorialspoint.com/angular8/angular8_directives.htm)

12. How can you create custom directive && while creating directive what are the files get updated

[Generate a Custom Angular Directive with the CLI | Pluralsight](https://www.pluralsight.com/guides/generating-a-custom-directive-with-the-cli)

13. What is the purpose using services in angular

[Angular Services - Why should we use Services in Angular? (unichrone.com)](https://unichrone.com/blog/angular/angular-service/#:~:text=Angular%20Services%20are%20used%20to%20complete%20a%20specific,all%20the%20operational%20tasks%20of%20the%20Angular%20components.)

14. What are the purpose of pipes

[Angular Pipes | Built-In Pipes & Custom Pipe with Example (ngdevelop.tech)](https://www.ngdevelop.tech/angular/pipes/#:~:text=Angular%20Pipes%20are%20used%20to%20transform%20data%20on,is%20like%20a%20filter%20in%20Angular%201%20%28AngularJS%29.)

15. what is the module in angular

[What is an Angular Module? - Rangle.io : Angular Training](https://angular-training-guide.rangle.io/modules/introduction)

16. How you will create production build

[How to bundle an Angular app for production? - GeeksforGeeks](https://www.geeksforgeeks.org/how-to-bundle-an-angular-app-for-production/#:~:text=The%20application%20builder%20in%20Angular%20uses%20the%20webpack,specifying%20the%20%E2%80%94%20configuration%3D%22production%22%20or%20the%20--prod%3D%22true%22%20option.)

17.JIT && AOT

[What is AOT and JIT Compiler in Angular ? - GeeksforGeeks](https://www.geeksforgeeks.org/what-is-aot-and-jit-compiler-in-angular/)

18.How you will pass data one to another component

[Passing data into Angular components with @Input - Ultimate Courses™](https://ultimatecourses.com/blog/passing-data-angular-2-components-input#:~:text=The%20first%20step%20to%20passing%20data%20into%20an,one%20component%20to%20another%20%28typically%20parent%20to%20child%29.)

19. Use of RXJS in angular

[Angular Observable Tutorial - TekTutorialsHub](https://www.tektutorialshub.com/angular/angular-observable-tutorial-using-rxjs/#:~:text=What%20is%20RxJS%201%20Reacting%20to%20an%20HTTP,uses%20observables%20to%20handle%20AJAX%20requests%20and%20responses.)

20. Difference ES5 && ES6:

[Differences between ES6 class and ES5 function constructors - GeeksforGeeks](https://www.geeksforgeeks.org/differences-between-es6-class-and-es5-function-constructors/#:~:text=Differences%20between%20ES6%20class%20and%20ES5%20function%20constructors,uses%20a%20new%20operator%20for%20object%20...%20)

21. Difference between one way && two way binding

[Difference between One Way and Two Way Data Binding with Example in Android Studio using Kotlin. (rrtutors.com)](https://rrtutors.com/tutorials/difference-between-one-way-and-two-way-data-binding-android#:~:text=Basic%20Differences%20between%20One%20Way%20and%20Two%20Way,Use%20can%20be%20in%20editable%20form%2Ffi%20...%20)

22. Have you did upgrade angular version

23. Lazy loading

[Angular Lazy-loading - javatpoint](https://www.javatpoint.com/angular-lazy-loading#:~:text=Angular%20Lazy-loading%20Lazy%20loading%20is%20a%20technology%20of,by%20the%20app%2C%20bundles%20are%20loaded%20as%20needed.)

24. Authguard Services in angular

[How To Add Authguard Service In Angular 5 (oodlestechnologies.com)](https://www.oodlestechnologies.com/blogs/How-To-Add-Authguard-Service-In-Angular-5/#:~:text=The%20Angular%20application%20can%20be%20guarded%20using%20authguard,logged%20into%20the%20application%20we%20use%20authguard%20service.)

25. Call, apply in js

26. Rest API

27. Closures

28. What kind of project you have worked

29. What is JSX

30.let, const, var

31. Object destructure - obj to array

32. Template url

33. Is image inline / block element - **Inline Element**

34. Async, Await

35. How you will upgrade yourself

**STERLING SOFTWARE:**

36.Lazy loading

37.Angular architecture

38.Payment gateway

39.Directives

40.Rxjs

**SIMPLE SOLVE:**

41.SwitchMap

42.JWT Token .io

43.Authorization, Authentication

44.Reactive programming

45.What is subject

46.How will you pass the data one page to another

47.Interceptor

**MOBIUS**

48.which file you will upload in server (Var file)

49.Lazy load

50.Data binding in angular

51.How angular will load?

52.

**DELOITTE:**

53.MVVM pattern

54.Content pattern

**RELEVENTZ:**

55.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

for (let i = 0; i < 5; i++) {

for (let j = i; j <= i; j++) {

console.log('\*-');

}

console.log('--');

}

56. Is JS async or sync?

57. Hoisting in js?

58. Single thread in js?

59. Let, var, const difference

60. Why TS

61. Any optimization in your code?

62. Lazy loading

63. How you debug your code

**TVSCAS**

64. ng-template

65. [‘x’,’y’] - [2,3]

66.Server side, Client Side rendering

**CBE NALAAS:**

67. String Reverse

68.Duplication removal

**Matrimony:** - 23/12/2021

-> Angular life cycle hooks

-> Different types of directives

-> Difference between and observable and promises :   
 1. Cancel the api promises they

(M - 5.02)

-> Pipes in angular

-> Storage in angular (Local, Session, index db or sql lite)

-> Module in angular

-> How can create production build

-> Git(just in) and AOT(A head of time)

-> What is the use Rxjs and where we can use it?

-> Without router how can we pass the data?

-> One way binding and two way binding

-> Angular upgrading or degrading

-> Lazy the module

-> Authguard services

-> Push notification

-> How you can update your self? Any blog or something?

-> How can you maintain your source code -> bit bucket or git

**LTIMind Tree:**

1. Routing through user interaction / url changes - How u will achieve this
2. Internalization - for multi-language
3. Karmajs & Jasmine with complex logic with service included something
4. Code to Compute π\piπ Using Leibniz's Formula
5. Get from Multiple Asynchronous api calls data & error management
6. Multi-device screen based responsive UI - how will do this?   
   -> Based on width media-query means how u will get that with runtime or ? (Exactly couldn’t remember ques)
7. Browser compatibility - how will things work in all browsers ?
8. How you will make your application as offline support in angular

### **Intellect Design:**

1. What is promises && Syntax for that
2. AOT & JIT
3. What is angular
4. Difference between Framework & Library
5. Authguard
6. Difference between Local-storage && Session-storage
7. Know about dynamic component loading?
8. Difference between Arrow function && normal function
9. Data passing parent to child
10. What is SPA?
11. Pipes in angular & how u will make custom pipes?
12. // const data = [ { category: 'A', value: 10 }, { category: 'B', value: 20 }, { category: 'A', value: 30 }, { category: 'B', value: 40 } ];

// Output: { A: 40, B: 60 }

1. // for(var i=0;i<5;i++){

// setTimeout(()=>{

// console.log(i)

// },1)

// }

**Wipro:(05/10/2024)**

1. Closure
2. Hoisting
3. Arrow functions, Norma function difference
4. Array destructure , obj destructure
5. Interceptor
6. Difference between Directive & component
7. Dependency injection
8. How you make performance optimization
9. About ng lifecycle hook
10. Kinds of directive
11. About SPA
12. Unit testing
13. SwitchMap, MergeMap
14. Viewchild, viewchildren
15. Ng-template, ng-container
16. Pseudo elements && pseudo classes
17. HTML - 5 semantic elements

**HCL: (06/ 10/ 24)**

1. Maximum char in word in js -   
   Ex: “PRIYANKA” O/P is A: 2
2. Recursive function
3. Interceptor
4. Error boundaries in angular   
   EX: if one component get error means another should run without an error. If header error means another side-panel, body, footer should work without crashing the application

**Paix Services:(09.01.2025)**

1. Throttle
2. Debounce
3. Pure function
4. Directive
5. Impure pipes
6. Box modal in css