

HTML / CSS / Angular / General FE Questions

HTML

1. DOM VS BOM
2. semantic element
3. inline vs block level element
4. web accesibility
5. img lazy loading
6. async vs defer tags

CSS

1. pseudo selector
2. BEM
3. positions
4. Grid/ flexbox
5. z-idex
6. Draw circle / triangle with css
7. Convert dekstop/ mobile view with css
8. media queries
9. Responsiveness

JavaScript

1. Hoisting
2. Debouncing / throttling
3. event bubbling
4. Callback
5. promise vs observable
6. Async await
7. ES6 syntaxes
8. Single page application with javaScript
9. let vs const vs var
10. EcmaScript
11. `new Array(3).toString()` // ""
12. `1 + '1'` // 11
13. `1 - '1'` // 0
14. <https://www.geeksforgeeks.org/trapping-rain-water/>
15. closure
16. `typeof class {}` / function / null / undefined
17. higher order function
18. event loop
19. Prtotype
20. call apply bind
21. prototype inheritance
22. optimise js,html app
23. `console.log('Hii')`
`fetch(/api).then()`
`setTimeout(function() {console.log('asda')}, 0)`
`console.log('bye')`
24. ES6 Features --- destructring, arrow, let const, async await, spread, Math.sign()
25. array vs object
26. functional programming
27. oops concepts

ANGULAR

1. Interceptor
2. lazy loading

3. Service. —> tree shaking
4. dependency injection in js
5. component level service vs global service @Injectable({provideIn: 'root'})
6. Pre loading
7. Rxjs Operators —> map/ filter/ switch map / retry
8. Subject / behavioural subject
9. Linting
10. Routing
11. server side rendering
12. cdk-virtual-scroll-viewport
13. cdkVirtualPort
14. Angular articture
15. How angular component can be used as class, directive
16. ng-contet vs ng-container vs ng-template
17. //Example:
import { Component, OnInit } from '@angular/core';

@Component({
 selector: '.mandatoryWithAsterisk, [mandatoryWithAsterisk]',
 template: `<ng-content></ng-content>`,
 styles: [':host::after{content: "**"; color: red }']
})
export class CustomMandatoryAsteriskClassComponent { }
18. Differential Loading
19. ngZone
20. viewState
21. in depth DB services
22. ngOnChnages
23. ngonInit
24. ngDoCheck
25. Router Gaurd -- can Load/ can Activate
26. sanitazation
27. Change Detection Stratergy - On Push
28. pipe/ impure
29. global error handling
30. Encapsulation
31. security -- RuteGaurd/Angular Template
32. Media Queries
33. State Management // store
34. Angular Universal
35. Angular element
36. ZoneJs
37. hooks
38. communication between components
39. NgRx/ Redux
40. Reducer
41. test cases jasmine/karma

Security / Performance

1. JWT
2. Mutifactor Authentication
3. CORS
4. cookie vs local storage vs session storage vs google authentication
5. http explenation —> headers —> user agent —> request token
6. Caching

FE Questions

1. Cross-browser compatibility
2. chrome - v8
3. mozilla - spiderMonkey
4. safari
5. What if i type google.com in vedio — how view renders (<https://medium.com/@maneesha.wijasinghe1/what-happens-when-you-type-an-url-in-the-browser-and-press-enter-bb0aa2449c1a>)
6. Debugging techniques/tools
7. dev tool lighthouse
8. web accessibility
9. microfront end/ mono repo

Build Tools

1. CICD Pipeline
2. DNS Server
3. Deployment procedure

Hoisting Problems

Input: nums = [2,7,11,15], target = 9

Output: [0,1]

Output: Because nums[0] + nums[1] == 9, we return [0, 1].

```
-----  
{"a":{"b":{"c":[1,2,3]}}}  
abvdf  
`{"a":{"b":{"c":[1,2,3]}}}`  
-----
```

```
var user1 = {  
  address:{  
    city: "Mumbai"  
  }  
}  
var user2 = user1  
user2.address.city = "Pune"  
// user1.address.city  
// user2.address.city  
-----
```

```
var user1 = {  
  address:{  
    city: "Mumbai"  
  }  
}  
var user2 = Object.assign({},user1)  
user2.address.city = "Pune"  
// user1.address.city  
// user2.address.city  
-----
```

```
var user1 = {  
  address:{  
    city: "Mumbai"  
  }  
}  
var user2 = {...user1}  
user2.address.city = "Pune"
```

```
// user1.address.city  
// user2.address.city
```

```
-----  
const a = 1  
a=2
```

```
-----  
const a = [1]  
a=[2]
```

```
-----  
const a = [1]  
a[0]=2  
-----
```

```
(function(){  
  var a = b = 3;  
})();  
console.log(a)  
console.log(b)
```