1.closure in depth
2.local storage session storage cookies
3.let/ var/ const
4.virtual DOM && dangerously.innerhtml,
5.Event Loop(how JS is converted to Async from sync single threaded)
6.subscribe in react
7.html symentic tags
8.createElement
9.React Fragments, Can we use classes in Fragment
-HOC functions in detail
10.binding in react (this.myfunction.bind())
11.how can we use media querry inside javascript
12.optimize web app(upload images size)
13.diff between [export default] and [export]
14.why reducers named reducer
15.hoisting- variable declared with name:xyz and function declared with same name, which one has more priority
16.REST api, how to test if an API is REST API
17.Create a login form, with some input fields and submit button , which will send inputed data to some api, and that data will be displayed in new component, and some other previously added info will also be there,
18.typeOf
19.memoization
20.Reference , uses of references.
21.LifeCycle react and getSnapshot
22.destructuring
23.ES6 new features
24.is setState() async or sync?
lexical scope vs closure

25.replacement of lifeCyclemethods with React hooks
26.useMemo vs useCallback
27.in depth about useEffect hook
28.how to iterate in a object, to get the key value pairs
29.for loop and for each loop, forEach vs map()
30.prop drilling
31.event bubbling and how to prevent it
32.Event capturing
33.strict mode in React
34.let vs var vs const **
35.what is binding in react components, and is it neccessary to bind?
36.functional vs class component react
37.what is prototype

38.lazy loading
39.callback hell
40.Promises vs async await
41.Promise.all
42.diff between npm and npx
43.Lazy Loading
44.object oriented javascript
45.webpack
46.Automation Unit Testing
47.call , apply, bind method in JS
48.props vs state
array.slice, splice, filter, map,shift, unshift, etc

50.super()
51.componentDidUpdate() , componentdidMount() in react hooks with useEffect()
52.why to use ARROW function, what are its extra advantages over usual functions.
53.pure components, shallow comparision vs deep comparision
54.which react hooks I have used
55.Which one you prefer class based components/ functional components
56.High order components
57.Disadvantage of using Refernce and when to use
58. portals in react

59.spread/ rest operator (deep/shallow)
60.how to remove duplicates from an arra
61.how to update the state (by setState)
62.react routes, switch
63.components of redux,
64.how to create custom components in React
65.state vs props ***
66.react limitations
67.pros and cons of react
Advantages of Func components over class based

CSS:
display, position, text-align, float left,transform, animation,
how to use gradient color in button,
section tag in HTMl
diff: diplay:flex and display:inline-flex
scss: functions/variable, mixins

---

what will be the output, if we do setState() more thn once in componentdiMount():

```
state({counter: 1})
componentDidMount(){
   this.setState({counter: this.state.counter + 1});
   this.setState({counter: this.state.counter + 2});
   this.setState({counter: this.state.counter + 5});
}
```

Ans: counter : 5

68.how redux thunk works

69. currying in JS.. sum(2)(3)(4) = 9

70. Axios

71. how to connect store to components.? asn: connect()

72. Browser Router, <Redirect to="">

73. Error boundaries in React (componentDidCatch , static getDerivedStateFromError())

74.lots of pre, post increment output quesns

75. how to connect store to components.? asn: connect()

how to empty an array

76: Generator Functions

77: dependencies vs dev-dependencies

78: is JS sync or async..

79: async vs promises

78: callback hell

79: can we pass negative value in settimeout

80: can we do push pop in const array

81: shallow vs deep copy

82: event bubbling, capturing

83: git cherrypick

84: arrow fn vs normal function

85: array built in methods: filter, find, etc

```
const arr = [1,2,2,3,4,4,5];
// Output of:-
// console.log(arr.filter(item => item != 2));
// [1,3,4,4,5]
// console.log(arr.find(item => item === 4));
// 1 true
console.log(arr.filter(item => item === 6));
// null
console.log(arr.find(item => item === 6));
// null
```

86: // Write the code to get the output as.

```
// [{1:1},{2:2},{3:3},{4:4},{5:5}]
// console.log(arr1.reduce(function(newArr,key,index){
//   newArr[index] = key;
//   return newArr
// },{}))
```

86: REST API’s , diff between POST and PUT

87: given a number, find the closest pallindrome number from it..

88: disadvantages of Vdom.

89: why setStateis async..whats the reason behind that

90: diff between them:

div p

Div > p

div ,p

91: controlled vs uncontrolled components