

generator
async await
multiple flavors
implement shouldPropsUpdated in functional
fetch or axios(<https://blog.logrocket.com/axios-or-fetch-api/>)
where we will store keys for aws for uploading file
background notification
get derived method
send params in navigation
reset the stack
currying
redux lifecycle
which is better class and functional
deep linking
1. proper flow of font file configurations

2. What is DOCTYPE and How many types of DOCTYPE

3. difference between html4 and html5

4. print a list (array of objects) using reusable component

5. how many type of position property

6. ref

7. hooks

8. context api with example

9. keyframes

10. memoization with example

11. Implement a modal which title, description and a button

12. Given 3 divs, should be displayed in same line and align in center

1. Promise functions which resolve after 5 secs
2. Created counter which prints from 1 to 5 with 5 secs gap in each output with help of above function
3. forEach and map difference
4. for in and for of difference
5. ES6 new function

6. React hooks and API

7. He asked me to rate myself in Node JS, React JS, Javascript & Mongo DB

8. New things in the New version of React

9. Main challenges which I have faced in any project in the last two yrs.

<https://github.com/lukehoban/es6features#reflect-api>

What will happen if we dont pass key in react

What is immutable and mutable

Primitive and Non Primitive Data Type

synthetic event

vanilla js

Reconciliation

[1]==[1]

undefined+12

12+null

"A" - 1

Three methods which can achieve in use Effect of lifecycle

Akahay saini

```
console.log('1')
```

```
setTimeout(()=>{console.log('2')},1000)
```

```
setTimeout(()=>{console.log('3')},0)
```

```
console.log('4')
```

A closure is the combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment).

Bridging

ES12

<https://ankitdeveloper.medium.com/100-most-asked-javascript-interview-questions-and-answers-part-1-1f151b14f0f7>

Extramake courier >>>

What we can use instead of axios and fetch

```
>>>>>>>>
```

```
console.log(a);
```

```
let a=1;
```

<https://blog.bitsrc.io/hoisting-in-modern-javascript-let-const-and-var-b290405adfa>

longestSubstr:

```
var longest=[];
```

```
function getsubstr(str){
  let longestLen=0;
  for(let i=0;i<=str.length-1;i++){
    for(let j=str.length;j>=0;j--){
      if(!checkDuplicate(str.slice(i,j))){
        longest.push(str.slice(i,j));
        if(str.slice(i,j).length>=longestLen){
          longestLen=str.slice(i,j).length;
        }
      }
    }
  }
  return longest.filter(item=>item.length==longestLen)
}
```

```
function checkDuplicate(data){
  const str=data.split("");
  const filtered=str.filter((alpha,index)=>str.indexOf(alpha)!=index);
  return filtered.length?true:false
}
```

```
console.log(getsubstr("Wabccggfhi"))
```

<https://medium.com/@markotroskot/debouncing-and-throttling-in-react-feb84d84bf28/>

Polyfills

ProtoTypal Inheritance

React js

- **What is react js and advantage of react js.**
- **What is different btw react js and other javascript framework like angularjs ,vue js**
- **What is component and state**
- **Library vs framework**
- **life cycle of component**
- **Redux hooks**
- **useCallback vs useMemo**
- **Event bubbling and how can we stop?**
- **Event capturing ?**
- **Server side Rendering?**
- **Virtual Dom vs Real Dom**
- **Error Bounderring**
- **Webpack or babel**

Redux

- **What is redux and lifecycle**
- **Redux thunk**
- **Redux vs context api**
- **Props drilling?**

Java script

- **What is javascript and async javascript**
- **First class function**
- **Normal function vs arrow function**
- **Call , apply and bind**
- **Callback and promise**
- **Deep copy vs shallow copy**
- **spread operator vs rest operator**

Node js

- **What is nodejs**

Console Question

// input : aaabbbbaaacccdd

// output: {a:3, b:3, c:3, d:2}

```
function getValueCount(str){
    const countObj={};
    let arr=str.split("");
    console.log(arr);
    for(let index=0;index<arr.length;index++){
        if(!countObj[arr[index]]){
            countObj[arr[index]]=0;
        }
        if(countObj.hasOwnProperty([arr[index]])){
            countObj[arr[index]]++
        }
    }
    return countObj;
}
console.log(getValueCount('121221234'))
```

```
var hello = function(){
    console.log(this);
}
var b = function(){
    console.log(this);
}
b();
}
console.log(hello());
-----
{
var a =10; let a =10;
console.log(a);
}
console.log(a);
-----
for(var i=0;i<3;i++){
    setTimeout(function(){alert(i);},1000+i);
}
```

#####

- 1: reverse string without using loop and inbuilt function
- 2: What is callback hell ? & why we use promise
- 3: event loop
- 4: create simple server using express
- 5: why use body-parse

Console

```
greetings("first");  
function greetings(a) {  
  console.log(a);  
}  
greetings("second");  
var greetings = function (a) {  
  console.log(a);  
};  
greetings("thirds");
```

#####