1. Difference between function(a,b){return a+b} and (a,b) => a+b
2. Which component you have designed by yourself, explain it how you are handling that.
3. What is the mean mapStateToProps and mapDispatchToProps?
4. Why we use redux, what is the use of redux.
5. Tell me the API methods which you have used.
6. What is the difference between PUT and Delete API methods?
7. Uses of API methods GET,POST,PUT,DELETE.
8. Which all type of collections you have used so far.
9. How to find the square of given array in JavaScript const x = [1,2,3,-3,-2,4] without using let and var.
10. How to pass unique records to automation from execl.
11. How to remove duplicate elements from a list or Arraylist.
12. Find the length of each word in a string and sort it from higher to lower order of length.

e.g.

Accomplishing 99.50% availability for Warranty Product with quality checks. Integrating Rest API using React, Javascript, Model factory. Automation integration Rest API Automation using newman. Hands-on experience on Git, Maven, Postman, SAP Gateway,BDD, Cucumber, TDD Accountable, Adoptable, Transparent, and integral about work

1. Find the emaloyee’s age where manager’s age is greater than employee’s age.
2. Difference between == and ===

null == undefined, null === undefined

1. var a = {name: "Mayank", surname: "Agarawal"} how will you print this in ES6.
2. What is the output of

(function(){

setTimeout(()=> console.log(1),2000);

console.log(2);

setTimeout(()=> console.log(3),0);

console.log(4);

})();

1. Difference between Flatten and Arrow Function.
2. What is Variable Hosting.
3. What is difference between useCallback,useMemo,useEffect?
4. What is the output of below program?

let x= {}, y = {name:"Ronny"},z = {name:"John"};

x[y] = {name:"Vivek"};

x[z] = {name:"Akki"};

x[y]

1. What is the difference between slice and splice.
2. var x = 10;

function printx() {

var x = 25;

console.log(this.x)

}

printx()

1. function runFunc(){

console.log("1" + 1);

console.log("A" - 1);

console.log(2 + "-2" + "2");

console.log("Hello" - "World" + 78);

console.log("Hello"+ "78");

}

runFunc()

1. What is NaN use?
2. What is hooks in react?
3. What is array de structuring?
4. What is the difference between let and var?
5. What is the difference between Local Storage and session Storage?
6. What is hooks in react, what are the hooks you have used in your development?
7. Can you explain the react lifecycle methods and how you use it in your development?
8. What is immediate initialization in javascript?
9. Const x=[11,2,3,4]

Y[x]=? And z[x]=?

1. How did you use or call child component inside the return function?
2. WAP to swap two numbers.
3. WAP to find the occurrence of elements in given array[1,2,3,4,1,2,1,5,6,7,1]
4. What is the difference between == and ===
5. What is the difference between state and props
6. What is the use of redux?
7. What is callback function?
8. What is Promise, why we use Promise?
9. What is refs in react?
10. What is pure component in react?
11. Use of useState and useEffect
12. What is th o/p: (‘0’+2),(1+0+’0’)
13. What is the o/p: 2==2,

2===2,

2==’2’,

2===’2’;

1. What is the mean of strict I javascript?
2. What is Lazy Loading?
3. What is Single Page Application?
4. What all the design pattern you have used?
5. What is Promise?
6. How you improve the performance of the application in react?
7. What is redux?
8. What is callback function?
9. What is virtual DOM?

**Spiralogics**

1. What is virtual DOM?
2. What all the hooks you have used in react?
3. What is Lazy loading in react?
4. How can you right given class component into function component?
5. import React from "react";
6. import ReactDOM from "react-dom";
7. import { Grid, Row, Col } from "react-flexbox-grid";
8. class App extends React.Component {
9. render() {
10. const force = 10;
11. const acceleration = 5;
12. const pressure = force / acceleration;
13. *//fsdasfd*
14. return (
15. <Grid>
16. <Row>
17. pressure:
18. <Col>{pressure}</Col>
19. </Row>
20. </Grid>
21. );
22. }
23. }
24. ReactDOM.render(<App />, document.getElementById("container"));
25. Did you face any security threats while integrating the application?
26. Did you used Redux thunk or redux saga?

**Atos:**

1. What is redux life cycle?
2. How to read the input from username field.
3. What is react lifecycle?
4. What is state?
5. What is props?
6. What is callback method?
7. WAP input string abcd output string bcda.
8. Suppose you are getting user details like address,DOB,age, when user successfully logged in you need to use these values in your home page, how will you use it?

**WIPRO:**

1. Find closest value to the given number, from the given array.

Given an array of sorted integers and find the closest value to the given number. The array may contain duplicate values and negative numbers.

Example : Array : 2,5,6,7,8,8,9 Target number : 5 Output : 5

Target number : 11 Output : 9

Target Number : 4 Output : 5

1. What is Virtual DOM
2. Difference between key and wraps in react?
3. Difference between call and apply?
4. Difference between let and var?
5. How many types of components in react?
6. What is typeof in JS?
7. What is redux?
8. What is the difference between functional component and class component in react?
9. Why we use setState?
10. What is the advantage of React, why we use react not angular?
11. What is flex in css
12. How many type of positions in css for a component.
13. What all type of features enabled in html5?

Kwglobal

1. What is fragment in react?
2. What is useRefs?
3. What is the difference between props and state?
4. How to pass value to child component
5. How to pass value to parent component?
6. What is higher order component?
7. What is box model in css
8. WAP input string “meeting” output String “emtenig”.
9. Difference between Map and Filter.

**CitusTech**

1. What is the difference between props and state?
2. How to pass value to child component
3. How to pass value to parent component?
4. Difference between let and var?
5. What is Virtual DOM, in which algorithm virtual DOM works.
6. What is typeof in JS?
7. What is redux?
8. What is the difference between functional component and class component in react?
9. What is react lifecycle?
10. Explain react hooks?
11. Create a function component that will display the message passed from props and will print a default message if not provided from props.
12. Reverse the input string in JS “Input: This is a paragraph”
13. What is the output of below program.

const foo = { key: "value" };

const foo\_copy = foo;

foo\_copy.key = null;

console.log(foo.key);

const obj = {

value: "foo",

printValue: () => {

console.log(this.value);

}

};

obj.printValue();

if (true) {

var var1 = "foo";

let var2 = "bar";

}

console.log(var1); // ?

console.log(var2); // ?

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

if (i === 2) continue;

console.log(i); // ?

}

if (true) {

var var1 = "foo";

let var2 = "bar";

if(false){

console.log(var2);

let var2 = 1250;

}

}

console.log(var1); // ?

console.log(var2); // ?

switch ("foo") {

case "foo":

console.log("foo");

case "bar":

console.log("bar");

}

x = 3;

console.log(x);

var x;

var x;

console.log(x);

x = 23;

let i = 0;

do {

i += 1;

console.log(i);

} while (i < 0);

console.log(1);

setTimeout(() => console.log(2), 0);

console.log(3);

1. What is closure in JavaScript?
2. What is lexical scope in JavaScript?
3. What is redux.
4. What do you mean by Single page application?
5. What is hosting in javascript?
6. Will it print console statement?

function testMeth(){

x = 33;

console.log(x);

var x;

}

1. What will be the output when I call makeFunc(x)

function testMeth(){

x = 33;

console.log(x);

makeFunc(x);

var x;

}

function newMeth(){

var x;

x = 33;

console.log(x);

makeFunc(x);

}

function makeFunc(var x) {

var a = x;

console.log(a);

var name = 'test';

function display() {

alert(name);

}

return displayName;

}

1. What will be the output when I call makeFunc(x)

function testMeth(){

x = 33;

console.log(x);

makeFunc(x);

var x;

}

function newMeth(){

let x;

var y=4;

let z = x + y;

x = 33;

console.log(x);

makeFunc(z);

}

function makeFunc(var ) {

var a = x;

console.log(a);

var name = 'test';

function display() {

alert(name);

}

return display;

}

1. What is the output

var a1 = 1;

var b1 = '1';

if(a1 == string){

alert(a1);

}elseif(b1 == string){

alert(b1);

}

1. What is the output

var a1 = 1;

var b1 = '1';

if(a1 == b1){

alert(a1);

}else if(b1 == a1){

alert(b1);

}

1. What is the output

var a1 = 1;

var b1 = '1';

if(a1 === b1){

alert(a1);

}else if(b1 === a1){

alert(b1);

}else{

console.log("No results");

}

1. How can we write default export is there any other way apart from this?

index.js

class Test extends React.Componenet{

render(){

<html>

<body>

<h3>Test Sample Comp</h3>

</body>

</html>

}

}

export default Test;

indexNew.js

import Test as {Test};

class NewComp extends React.Component

{

NewComp(){

this.super();

}

render(){

return "<html><body>TestNewComp</body></html>"

}

}

1. What is function component and class component?
2. What is react lifecycle method.
3. Example of Closure(Function inside the function).

function makeFunc() {

var name = 'test';

function display() {

alert(name);

}

return display;

}

1. In above example var name scope is lexical scope, if we move out var name outside this function, it will throw the error.
2. What is the output

var a1 = 'Ram';

var b1 = "Ram";

if(a1 typeof string){

alert(a1);

}else if(b1 typeof string){

alert(b1);

}

1. What is the output

var a1 = 'R';

var b1 = "R";

if(a1 typeof string){

alert(a1);

}if(b1 typeof string){

alert(b1);

}

**Accenture:**

1. Difference between useeffect and useLayouteffect?
2. Difference between async and defer
3. How you optimize the performance of webpage in your application.
4. What is Continuous Integration.
5. What is Continuous deployment.
6. How Jenkins pipeline get to know need to trigger automatically?
7. What is devops?
8. What is the difference between Library(React) and Framework(Angular).
9. Why do we need Jenkins.
10. How you optimize the performance of your application?
11. What all the features react provided?
12. What is virtual DOM.
13. What is the difference between State and Props.
14. Difference between one way and two way binding and which one is better?
15. What is watcher?

**GrandPrade:-**

1. What is Box Model in CSS.
2. What is FlexBox in CSS.
3. What is difference between Id and class in CSS.
4. What is !important in CSS.
5. What is preprocessor in CSS.
6. What is mean by cascading in CSS?
7. What are the new feature of CSS3?
8. Difference between let, var and cost?
9. What type of scope present in javascript?
10. What all the new features ES6 has?
11. What is inheritance in JS.
12. What is the advantage of JS over other languages?
13. How you will do cross browser testing(Cors policy and JsonView)
14. What is prototype inheritance in javascript, how can we achieve?
15. How many ways you can write a function in JS?
16. Is javascript asynchronous or synchronous?
17. What is difference between local storage and session storage?
18. What do you mean by synchronous?
19. How many type of variable in JS
20. What is NaN in javascript?
21. What is Boolean in JS?
22. Difference between == and ===?
23. React advantage
24. Why we are using react.
25. What is react lifecycle?
26. How can we improve the performance of a web application?
27. What is store?
28. Difference between state and props?
29. Flow of component rendering?
30. Did you use Jest to right unit test cases?
31. What is rebase,cherrypick,merge?
32. Why we use Jenkins?

**DSV:**

1. What is Event bubbling?
2. What all the design pattern you have used?
3. What is the difference between const and static in Java
4. What is the difference between local storage, session storage and cookies.
5. How you are handling cross browser functionality in react?
6. How you optimize the performance of your website?
7. What is Lazy Loading?
8. What lazy loading do in real?
9. What is difference between let and const.

**AXTRIA:**

1. How you are deploying your builds in production, who configured the Jenkins pipeline for that?
2. Can you please merge two sorted list without using built in function?
3. What is redux?
4. Can you please explain the redux lifecycle?
5. What all the design pattern you have used in react?
6. How do you decide to need to make common component or generic?
7. Are you using async call or sync call in your application?

**DEVKRAFT:**

1. WAP to print a list of employee name and salary.
2. Convert below class component into function component.
3. import React from "react";
4. import "./styles.css";
5. class App extends React.Component {
6. constructor() {
7. super();
8. this.state = {
9. someVar: true
10. };
11. }
12. componentDidMount() {
13. console.log("");
14. }
15. componentDidUpdate(prevProps, prevState) {
16. console.log("");
17. if (prevState.someVar !== this.state.someVar) {
18. console.log("");
19. }
20. }
21. componentWillUnmount() {
22. console.log("");
23. }
24. shouldComponentUpdate(prevProps, prevState) {
25. if (
26. prevState.someVar !== this.state.someVar &&
27. this.props.someProp !== prevProps.someProp
28. ) {
29. return false;
30. }
31. return true;
32. }
33. render() {
34. return (
35. <div className="App">
36. <h1>Hello CodeSandbox</h1>
37. <h2>Edit to see some magic happen!</h2>
38. </div>
39. );
40. }
41. }
42. export default App;

**TicketMaster**

1. What is Redux?
2. Can we use only local storage and session storage not redux.
3. Difference between let and var
4. Difference between callback() and Promise().
5. Is Javascript single threaded or multi-threaded.
6. If we open multiple tabs for same website, is JS waits for other tabs to perform the operation or not
7. Which all the version you are using in React and the difference between them.
8. What all the lifecycle methods of react.
9. When I call setState which lifecycle’s methods will hit.
10. WAP

\*\*\*\*

\*\*\*

\*\*

\*

**No Broker:**

1. How you are calling your API’s?
2. Can you explain redux lifecycle?
3. Why we are using hooks, what are the advantage of react hooks over lifecycle method?
4. What is the difference between call and apply?
5. What is the difference between SCSS and SASS?
6. WAP merge two sorted array?
7. What is IIFE?

**SHL:**

1. What is the output of below program?

function test() {

console.log(a);

var a = 10;

console.log(a);

}

test();

console.log(a);

function test() {

{

let b = 10;

}

console.log(b);

}

test();

1. How do you clear the array e.g. a[1,2,3]?
2. What all the methods you have used on Array like slice sort?
3. How do you apply the other methods top of a.sort?
4. What is the difference between event .prevent default and event stop propagation?
5. What is Event Propagation(Event Bubbling)?
6. What is Promise?
7. How Promise works wen you are fetching any API?
8. How to handle PromiseReject?
9. Can you fetch response without suing async and await?
10. What is the significance of bind method?
11. What is the difference between slice and splice
12. Which lifecycle method you will use to re-render the component?
13. Difference between shallow and mount in test cases?
14. Difference between arrow function and normal function
15. Which framework you are using for CSS?
16. What is the difference pure component on normal companent?
17. What is closure?
18. What is the significance of closure, why we use closure?
19. How do you use Prmoise if you don’t use async or await

**Axtria**

1. What is wrong with below code?

this.setState({

counter: this.state.counter + this.props.increment,

});

1. How can we correct this.

This.state.message =”Hello world”;

1. What is event drilling?
2. Why we use Refs in React?
3. What is the advantage of class component over functional component?
4. Are you using any performance optimization tool for your application?
5. What is difference between React JS and Angular JS
6. What is the difference between JavaScript and TypeScript?
7. What all the hooks you are using in your application?
8. Why we use Use State?
9. Why we don’t update State in React , Generally?
10. Why we use useState hook?
11. What is Redux?
12. What is controlled component?
13. Difference between presentational and container component?
14. How redux is helpful for maintaining the state?
15. Why we use three dots in our react code?
16. How to initialize the constructor in react?
17. Explain react life cycle?
18. Why we use refs?
19. Why we do binding or significance of bind() method?
20. What is event handler?

**const a='john doe'**

**for(let i=0;i<a.legth;i++)**

**{**

**for(let j=1;j<a.legth();j++)**

**{**

**if(a[i]==a[j])**

**{**

**console.log("Print chracter", a[i]);**

**}**

**}**

**}**

**How to write simple map on aaray objects or**

**purchasedProducts [**

**{**

**orderId:1,**

**productId:1,**

**userId:1,**

**qty : 2,**

**date:"02/11/2021"**

**discount: 10**

**}**

**]**

**Fabric**

function getA() {

console.log('a')

}

function getB() {

console.log('b')

}

setTimeout(getA, 0)

getB()

let c = 2

console.log(c)

/\*

b

a

2

\*/

/\*

Input

const foods = ['apple', 'orange', ['grapes','banana',['mangos','kiwi']]]

Expected out

['apple','orange','grapes','banana','mangos','kiwi']

Q: flatten nested array

\*/

**Maveric:**

1. Explain react lifecycle hooks?
2. Write a program to find the occurrence of a first repeated character in given string ‘John Doe’?
3. What is the difference between React DOM and React?
4. Why we write<doctype> at the start of any HTML file?
5. What is the mean of margin 1px,2px,3px,4px?
6. How axios works internally?

**Gobant:**

1. Difference between CSS and SCSS.
2. Difference between let and var and const?
3. Why we are using hooks or why hooks are introduced?
4. Can we use hooks inside the class?
5. How to align child div into center, what property you will define?
6. What is semantic elements?
7. Difference between local storage and session storage?

**Cognizant:**

1. What are semantic elements?
2. What are the from elements in HTML?
3. What is synthetic event in react?
4. What are error boundaries in react?
5. How to avoid re-rendering of component in class as well as function component?
6. When we are calling any API which lifecycle method will call 1st?
7. What is the difference between slice, splice and split?
8. Difference between let and const
9. What all the flex box properties?
10. What are the positions in CSS?
11. What we need to do when we are converting any simple react project into redux?
12. Difference between props and state?
13. What is arrow function, why we use that?
14. What all the ES6 features?

**Energy Exampler:-**

1. How to find all palindrome substring from the given String?

Input:- ‘ababaca’

Output: a,b,ab,ba,aba,bab

1. What all the property box has?
2. What all are the position for a div you have used?

**IBM:**

1. WAP to remove duplicate element from array.
2. Difference between slice and splice?
3. How do you identify which lifecycle method is calling and when in functional component?
4. Which method you will use to achieve component did munt and component will mount?
5. Did you write your custom hook?
6. Difference between Shallow copy and Deep copy?
7. Significance of Call, Apply and Bind in JS?
8. Difference between Arrow function and Regular Function?
9. Async/Await
10. What is observer Pattern or Command Pattern?

**HSBC:**

1. What is Memo in react.
2. What is refs in react
3. How to resolve re-currying rendering in React.
4. How to handle component while browser re-sizing?
5. Did you use any other State Mgmt tool apart form React.
6. Difference between create element and clone element
7. How to avoid prop-drilling?

**NICE:**

1. What is Memo in react.
2. What is pure function in react
3. What is controlled component and uncontrolled component in react.
4. What is keys in react?
5. How to improve the performance of our react application?
6. What are the advantage and limitation of React?
7. How your real DOM get to know that virtual DOM is updating.

**PEAK**

1. Why we are not able to see HTML and CSS code if we do Ctrl+P?
2. What is single page application
3. How do you dynamic search from search box when you type f it should return all the results in dropdown like google search.(You need to call the api without clicking on search button)
4. Which TDD approach you are using to write the test cases.(property based or functional)
5. What is server-side rendering?
6. What is the benefit of TDD
7. Outout:-

async function handler() {

const promise = new Promise((res, rej) => setTimeout(res('some error'), 1000))

try {

return promise;

} catch(err) {

console.log('inside err', err)

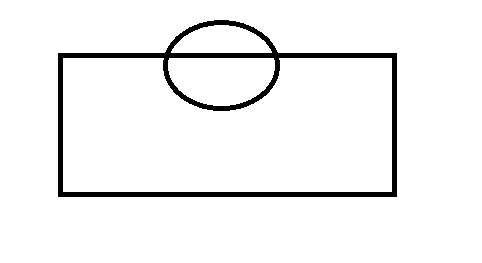
}

}

handler().then(data => console.log('data', data)).catch(err => console.log('outside err', err))

**authbridge**

1. Why we change the file format from JSX to JS?
2. Why browser display .html page only not other JS page or .jsx pages?
3. How to avoid re-rendering of components in react?
4. How to improve performance of our react application?
5. What is closure in JS(A part from function inside the function)?
6. What is the difference between error handling and error boundary?
7. What all the flex box property a child element can have?
8. Difference between session storage and local storage?(F12->Application->Storage)
9. How to achieve inheritance in JS?
10. Difference between windows.get() and Document.get()?
11. Why we define synthetic tags in HTML page?(Header,Div, footer). These tags help in SEO.
12. What is the difference between display none and display 0px
13. What are the required prop types?(boolean, function, array, object)
14. Can you write a code for this shape



1. function func() {

return x;

var x = 5;

function x() {};

var x = 10;

}

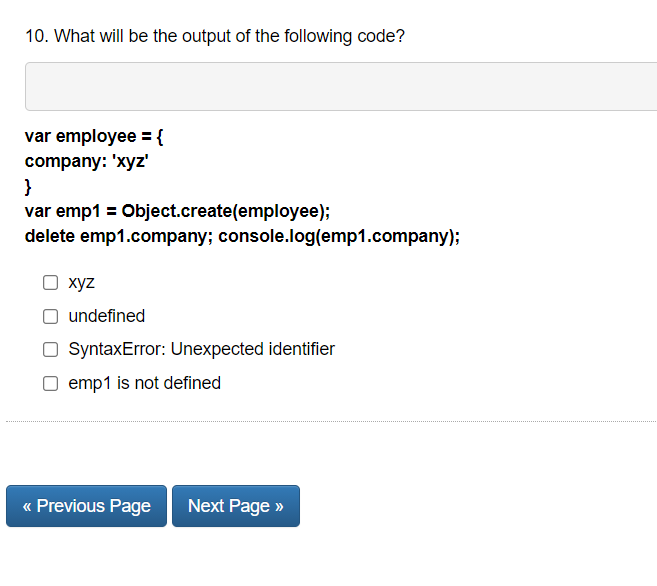
console.log(func());

What is the output and why?

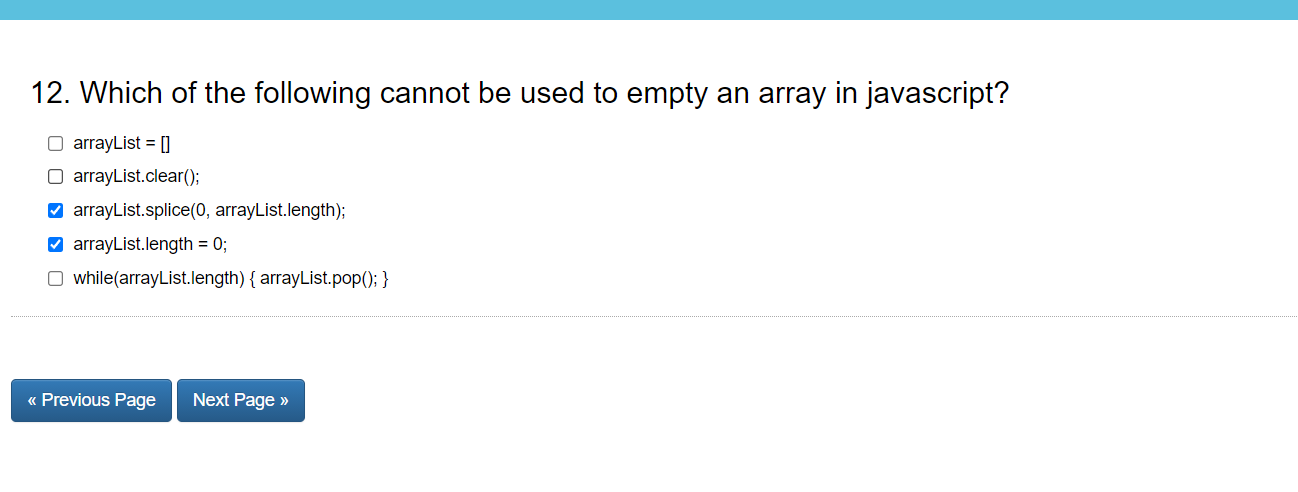
**Ztech**

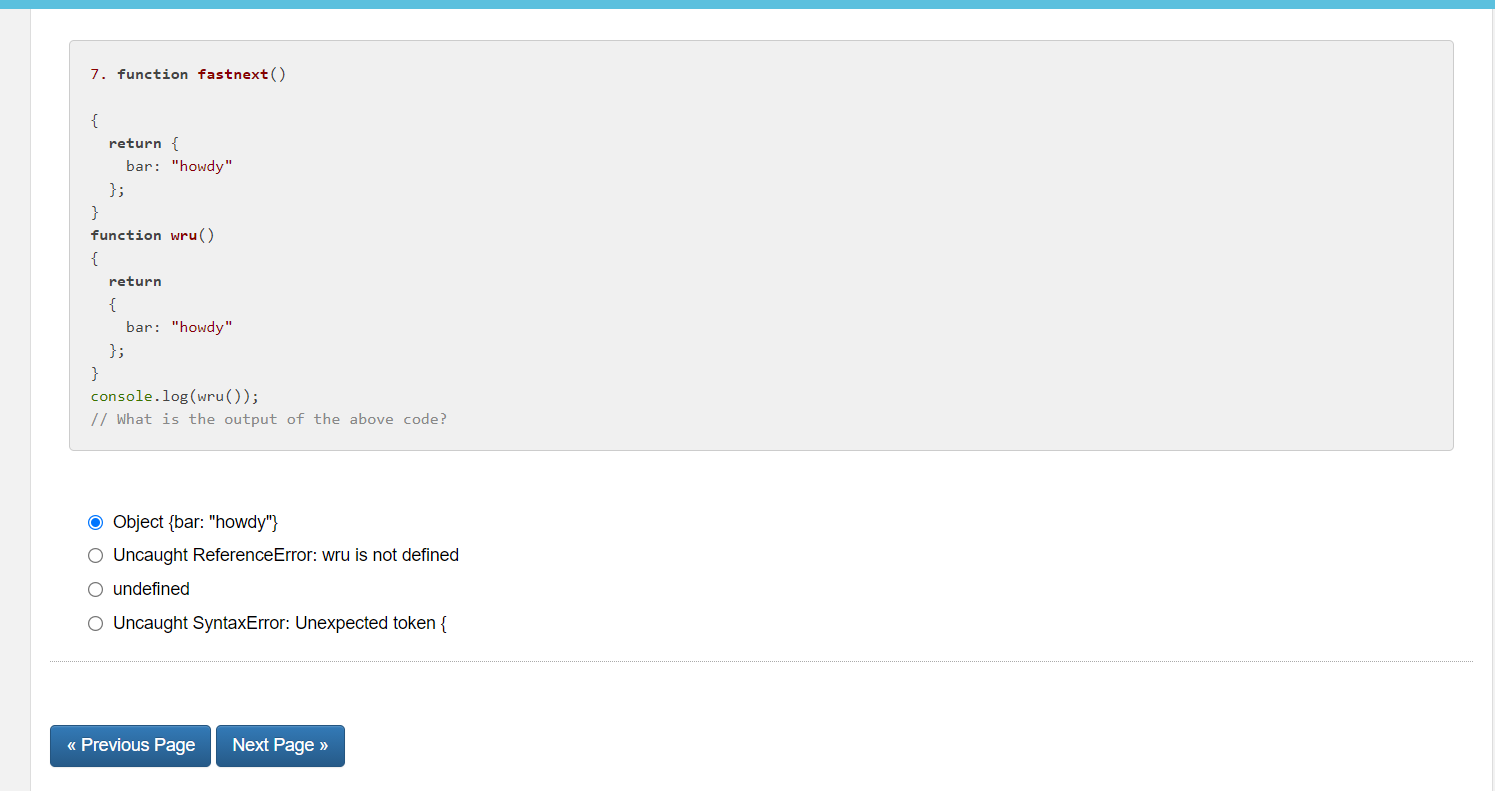
1. What is react Pure component?
2. How do you pass the value from redux store to your component?
3. How do you modify the value in redux store or to pass modified value to redux store?
4. What is reducer?
5. What is pure function?(Reducer is an example of pure function)
6. Connect method redux is an example of pure function.
7. How to avoid prop drilling? (Using redux and context api)
8. What is higher order component?

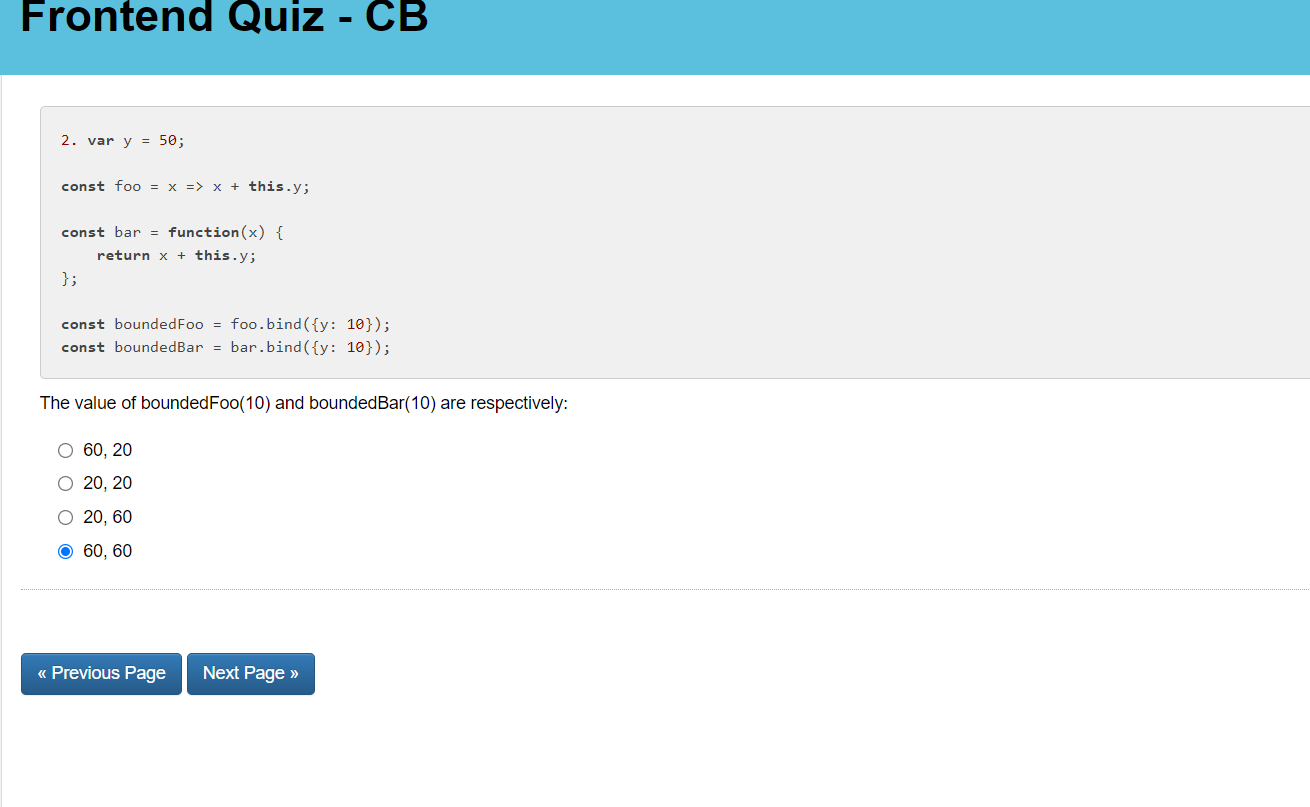
Online test

****

****

****

****

****

**Nickelfox**

1. Why we use middleware?(Redux thunk or saga)(these middleware wait for the response and make our page running, it async the reponse)
2. What is the difference between map and foreach?
3. What is the difference between map and forin?
4. How can we delete the element from object?
5. What is the output before uncommenting hosting and after uncommenting and why?

**// solve and explane**

**var myname = "global";**

**let hoisting = () => {**

**console.log(myname);**

**var myname = "local";**

**console.log(myname);**

**};**

**// hoisting();**

1. What is the output before uncommenting closure and after uncommenting and why?

**// Closure**

**const closure = () => {**

**for (var i = 0; i < 4; i++) {**

**setTimeout(function () {**

**console.log(i);**

**}, 0);**

**}**

**};**

**// closure();**

1. What is the output before uncommenting eventloop and after uncommenting and why?

**// Event Loop**

**const eventLoop = () => {**

**console.log(1);**

**setTimeout(() => console.log(2), 0);**

**Promise.resolve(3).then((res) => console.log(res));**

**console.log(4);**

**};**

**// eventLoop();**

1. What is the output before uncommenting asyncEvent and after uncommenting and why?

**// solve**

**// Async Event Loop**

**const asyncEvent = async () => {**

**console.log(1);**

**setTimeout(() => console.log(2), 0);**

**await Promise.resolve(3).then((res) => console.log(res));**

**console.log(4);**

**};**

**// asyncEvent();**

1. How to remove duplicates from array?

**// Unique array**

**// => [1, 2, 3, 4]**

**const unique = [1, 2, 1, 1, 2, 1, 3, 4, 1];**

**// console.log();**

1. How to merge two array objects?

**// Merge array of objects**

**// =>  { a: 1, b: 2, c: 3 }**

**const merge = [{ a: 1 }, { b: 2 }, { c: 3 }];**

**function mergeObject(list) {}**

**// console.log(mergeObject(merge));**

1. How to remove one property from object?

**// Remove Property**

**const removeProperty = {**

**foo: "bar",**

**useful: 1,**

**useless: 2**

**};**

**function removeProp(obj, propName) {}**

**// console.log(removeProp(removeProperty, 'foo'));**

1. How to cover array into object?

// Array to Object

// => {1: 2, 2: 1, 3: 1, a: 2, b: 1}

Output: const array = ["a", 1, "a", "b", 1, 2, 3];

function calculateDublicates(list) {

list.reduce((acc, next) => {

acc[next] = acc[next] ? acc[next]++ : 1;

// if (acc[next]) {

// acc[next]++;

// } else acc[next] = 1;

// console.log(acc, next);

return acc;

}, {});

}

console.log(calculateDublicates(array));

1. How to generate a dynamic html table with rows and columns?

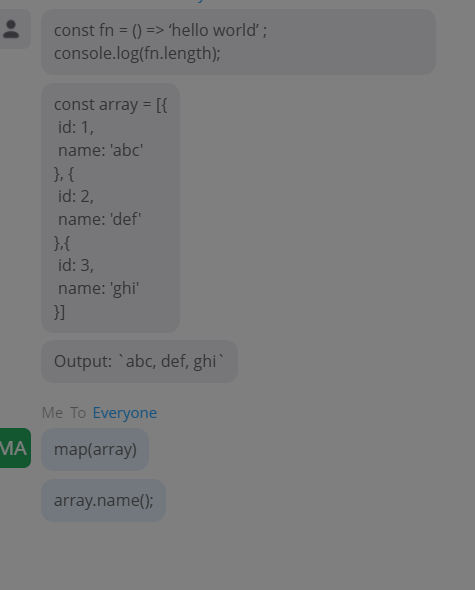
[**invansys**](mailto:amanpreet@invansys.com)

1. What is the difference between rest operator and spread operator?
2. Difference between map and filter?
3. What is use Selector?
4. Is setState synchronous or asynchronous method?
5. What is the output of below statement?

Const fn=()=>’hello world’

Console.log(fn.legth);

1. Why we use middleware, what is the need to use them?
2. What is the use of useSelector?
3. Reduce function?
4. Write the code for desired output

****

**Globant:-**

1. What is the difference between rest operator and spread operator?
2. How to create object in Javascript?object.create(), Object Literals
3. What is the difference between function and arrow function?
4. How hosting works on function level?
5. What is aysnc and await, why we use await?
6. Why we use await, if we didn’t get the response what will happen in case of await?
7. Call, apply, bind in javascript?
8. HTML5 tags
9. Can we see virtual DOM and Real DOM?
10. How did you implemented session storage ans local storage?
11. Difference between foreach and map?
12. Difference between filter and reduce?
13. Can we use variable of script A inside Script B, if yes how can we do that?

**// script A**

**let i=0;**

**// scrip B**

**console.log(i);**

1. How can we create multi argument function?
2. Difference with flex box and box Model?
3. How can we avoid two same class CSS into a component or tag?
4. How can we write component did mount().compnent did update(), compnenet will mount using useeffect?
5. Explain react lifecycle
6. Difference between virtual DOM and real DOM
7. Difference between context API and redux?
8. When you are writing in a text box it should write below the text box and if I click on reset button after text it should clear those text.

**Output questions:-**

1. **(function(){**

**console.log(1);**

**setTimeout(function(){console.log(2)},1000);**

**setTimeout(function(){console.log(3)},0);**

**console.log(4);**

**})();**

**Ans: 1,4,3,2**

1. **let a=b=018-017;**

**console.log(1);**

**console.log(b+2);**

**Ans:1,5 b values is 3 here.**

1. **var a=100;**

**function test(){**

**if(false)**

**{**

**var a=199;**

**}**

**document.write(a);**

**}**

**test();**

**Ans Un defined bcos none of the a value is accessable**

All You Need to Know to Get Started With JavaScript Promises 🙋🔥  
Many people struggle with understanding how Promises work, so I will try to explain them as simply as I can.🎓  
  
👉 Imagine you are a little kid, your mommy promises you that she'll buy you a new computer next week.🤵  
You don't know if you will get that computer until next week.🤔  
Mom can either buy you a new computer, or she doesn't, because you didn't clean your room, and she isn't happy!🛏️  
  
That is a promise! A promise has 3 states:  
🔶Pending: You don't know if you will get that computer.  
🔶Fulfilled: Mom is happy, she buys you a new computer.  
🔶Rejected: Mom is unhappy, she doesn't buy you a computer.  
  
🏁 Creating and Using A Promise 🏁  
Firstly, we use a constructor to create a Promise object in [#javascript](https://www.linkedin.com/feed/hashtag/?keywords=javascript&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6812990047718133760), It takes two parameters, one for success (resolve) and one for fail (reject),  
  
Finally, there will be a condition. If the condition is met, the Promise will be resolved, otherwise, it will be rejected:  
  
✔️ .then( ) - for resolved:  
If the Promise success, then something will happen next depends on what we do with the resolved Promise  
  
❌ .catch( ) - for rejected:  
What if the Promise fails? it will jump to the catch( ), therefore we need to use the catch( ) method.