**Ask Questions 15m**

**1. You are rendering a list with React when this warning appears in the console: "Warning: Each child in a list should have a unique 'key' prop." How do you fix this issue?**

A. Pass the name of each item as its key.

B. Add a key prop with the same value to each item the list.

C. Clear the console warnings.

D. When iterating over the list items, add a unique property to each list item.

**2. In this component, how do you display whether the user was logged in or not?**

***render() {***

***const isLoggedIn = this.state.isLoggedIn;***

***return (***

***<div>***

***The user is:***

***</div>***

***);***

***}***

A. The user is loggedIn ? logged in : not logged in.

B. Write a function to check the login status.

C. The user is {isLoggedIn = "no"}.

D. The user is {isLoggedIn ? "logged in." : "not logged in"}.

**3. Filter, Map and Reduce operation is way to \_\_\_\_\_\_\_\_\_\_\_\_\_\_ .**

A. Un-compress and expand data.

B. Create new immutable dataset.

C. Crunch and analyze the data

D. All of them (bir yazı okumak istenirse)

**4. What is the first file loaded by the browser in a basic React project?**

A. src/App.js

B. src/index.js

C. public/index.html

D. public/manifest.json

5. **What is the “key” prop?**

A. “Key” prop is just there to look pretty and there is no benefit whatsoever

B. “Key” prop is a way for React to identify a newly added item in a list and compare during the “diffing” algorithm

C. It is one of the attributes in Javascript and HTML

D. All I know is that it is NOT commonly used in array

6. **How do you handle passing through the component tree without having to pass props down manually at every level?**

https://reactjs.org/docs/context.html

Context provides a way to pass data through the component tree without having to pass props down manually at every level.

A. React Send

B. React Pinpoint

C. React Router

D. React Context

**7. Why might you use useReducer over useState in a React component?**

A. when you want to replace Redux

B. when you need to manage more complex state in an app

https://dev.to/spukas/3-reasons-to-usereducer-over-usestate-43ad

C. when you want to improve performance

D. when you want to break your production app

**8. Which props from the props object is available to the component with the following syntax?**

<Message {...props}/>

A. any that have not changed

B. child props

C. any that have changed

D. all of them

9. **What is the difference between the click behaviors of these two buttons (assuming that this.handleClick is bound correctly)?**

X. <button onClick="{this.handleClick}>Click Me</button>"

Y. <button onClick="{event => this.handleClick(event)}}>Click Me</button>"

A. Button X will not have access to the event object on click of the button

B. Button X will not fire the handler this.handleClick successful we can not say these

C. Button Y will not fire the handler this.handleClick successfully

D. There is no difference

10. **How do you destructure the properties that are sent to the Dish component?**

function Dish(props) {

return (

<h1>

{props.name} {props.cookingTime}

</h1>

);

}

A. function Dish ([name, cookingTime]) { return <h1>{name} {cookingTime}</h1>; }

B. function Dish(...props) { return <h1>{name} {cookingTime}</h1>; }

C. function Dish(props) { return <h1>{name} {cookingTime}</h1>; }

D. function Dish({name, cookingTime}) { return <h1>{name} {cookingTime}</h1>; }

**11. Which attribute do you use to replace innerHTML in the browser DOM?**

<https://quick-adviser.com/which-attribute-do-you-use-to-replace-innerhtml/>

dangerouslySetInnerHTML is an attribute under DOM elements in React. According to the official documentation, dangerouslySetInnerHTML is React’s replacement for using innerHTML in the browser DOM.

A. injectHTML

B. dangerouslySetInnerHTML

C. weirdSetInnerHTML

D. strangeHTML

**12. What is this pattern called?**

const [count, setCount] = useState(0);

A. object destructuring

B. spread operating

C. array destructuring

D. code pushing

13. **What is wrong with this code?**

const MyComponent = ({ names }) => (

<h1>Hello</h1>

<p>Hello again</p>

);

A. React does not allow components to return more than one element.

B. React components cannot be defined using functions. C. The component needs to use the return keyword.

D. String literals must be surrounded by quotes

**https://www.freecodecamp.org/news/quick-guide-to-understanding-and-creating-reactjs-apps-8457ee8f7123/Basic Folder Structure Explained. Basic Folder Structure Explained**

When you created the project, you would have noticed that it created a bunch of files. Here I will list out some of the important files and folders that you should be aware of .

**package.json:** This File has the list of node dependencies which are needed.

**public/index.html:** When the application starts this is the first page that is loaded. This will be the only html file in the entire application since React is generally Written using JSX which I will cover later. Also, this file has a line of code <div id=”root”></div>. This line is very significant since all the application components are loaded into this div.

**src/index.js:** This is the javascript file corresponding to index.html. This file has the following line of code which is very significant. ReactDOM.render(<App />, document.getElementById(‘root’));

The above line of code is telling that App Component ( will cover App Component soon) has to be loaded into an html element with id root. This is nothing but the div element present in index.html.

**src/index.css:** The CSS file corresponding to index.js.

**src/App.js :** This is the file for App Component. App Component is the main component in React which acts as a container for all other components.

**src/App.css :** This is the CSS file corresponding to App Component

**build:** This is the folder where the built files are stored. React Apps can be developed using either JSX, or normal JavaScript itself, but using JSX definitely makes things easier to code for the developer :). But browsers do not understand JSX. So JSX needs to be converted into javascript before deploying. These converted files are stored in the build folder after bundling and minification. In order to see the build folder Run the following command

npm run build

**Interview Questions 15m**

**1. What are React Hooks?**

**2. Why should component names start with capital letter?**

**3. What is Suspense component?**

**4. What is React context vs React redux?**

**5. What are controlled components?**

**6. How Virtual-DOM is more efficient than Dirty checking?**

Coding Challenge 20m

Coding Challenge: JS-CC-025 Roman Numearls

Project Skeleton

004 - Random User App (folder)

|

|----readme.md # Given to the students (Definition of the project)

SOLUTION

├── public

│ └── index.html

├── src

│ ├── assets.js

│ │ └── [images]

│ ├── App.js

│ ├── App.css

│ ├── index.js

│ └── index.css

├── package.json

└── yarn.lock

Expected Outcome

Project 004 Snapshot

