

Which of the examples below are valid types for Elements? Select all that apply.

- type: "button"

This is an example of how React specifies a 'type' attribute when performing a tree transformation with simple DOM nodes.

- type: DeleteButton

React would identify a component named DeleteButton as a function and ask that component what element it renders to, with the given props.

Where DeleteButton is a React component.

- type: "div"

This is an example of how React specifies a type attribute when performing a tree transformation with simple DOM nodes.

Which of the statements below clearly states the definitions of Containment and Specialization? Select all that apply.

- Containment refers to the fact that some components don't know their children ahead of time. And can also be described as generic boxes, like a Dialog or Alert.
- Specialization defines components as being "special cases" of other components. For example, the ConfirmationDialog is a special case of Dialog.

Which of the following operations does the React.cloneElementAPI enable a parent to perform? Select all that apply.

- Add to children properties
- Extend the functionality of children components
- Modify children properties

Props in React are immutable objects, so once React.cloneElement has created a copy of the element it can then:

- Modify the children's properties in the copy.
- Add to the children's properties in the copy.
- Extend the functionality of the children components.

A simple welcome screen for Little Lemon restaurant has been built, where users are able to sign up or login, depending on whether they have an account or not. The App component renders two buttons and uses the Button component for the sign up and the LoginButton component for log in. Both buttons make use of the onClick handler, to show an alert about the intended action when they are pressed.

The same alert message that has been provided for Sign up, is used on the LoginButton component, hence overriding the onClick handler that the LoginButton already defines internally.

What would the message of the alert be when the Sign up button is clicked?

Logging in. Even though the onClickprop in the LoginButtoncomponent is overridden, its implementation prevents that overriding from happening, due to the order of the spread operator.

A higher order component (HOC), is an advanced pattern that emerges from React's compositional nature. What are some of the benefits of using a HOC in your solution?

- It enhances or extends the capabilities of the component provided.
- You can define logic in a single place, share it across many components, and keep them unchanged and stateless.

A HOC transforms a component into another component. In other words, it enhances or extends the capabilities of the component provided.

When defining a HOC, such as `withMousePosition`, why is the `with` part of the HOC name a general convention recommended by React?

It expresses the enhancing nature of the technique.

The `with` part of the HOC name is a general convention recommended by React, as it expresses the enhancing nature of the technique, like providing a component ‘with’ something else.

What is the result of the return statement of the `DataFetcher` component?

The component returns the result of calling the render function.

The `DataFetcher` component returns the result of calling the render function and has no other rendering business.

Which of the following statements are true about Jest and React Testing Library tool? Select all that apply.

- Jest is a JavaScript test runner that lets you access an artificial DOM called jsdom. While jsdom is only an approximation of how the browser works, it is often good enough for testing React components.
- React Testing Library is a set of utilities that let you test React components without relying on their implementation details. It is designed to fulfill all testing best practices out of the box, so that you are able to focus on the business logic your tests need to run assertions on.

When writing a part of your test to locate the range input and fill it with a value, which functions do you need to use?

- `fireEvent.change()`

To fill the input and update the state with a value, you can use the `fireEvent.change()` utility from React Testing Library.

- `screen.getByLabelText`

`Screen.getByLabelText` asks the root document to find a label tag whose text contains a specified text value and then return the input element associated with that label.