Q.1 What are hooks in react? how to identify hooks?

React Hooks are a feature in React that allows developers to use state and other React features without writing a class component. Hooks are functions that let developers "hook into" the React state and lifecycle methods from a functional component.

Q.2 Explain useState Hook & what can you achieve with it?

The useState() hook is a function that returns an array containing two elements: the current state and a function to update the state. The useState() hook is used to add state to a functional component.

```
import React, { useState } from 'react';

function Example() {

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

return (

<div>

<p>You clicked {count} times</p>

<button onClick={() => setCount(count + 1)}>

Click me

</button>

</div>
```

);

}

Q.3 How to pass data from one component to another component

For passing the data from the child component to the parent component, we have to create a callback function in the parent component and then pass the callback function to the child component as a prop. This callback function will retrieve the data from the child component

Q.4 What is the significance of the "key" prop in React lists, and why is it important to use it correctly?

React uses the key prop create a relationship between the component and the DOM element. The library uses this relationship to determine whether or not the component should be re-rendered. It is not recommended to use the index of the array as the key prop if you know the array will not be static.

Q.5 What is the significance of using "setState" instead of modifying state directly in React?

The setState() method enqueues all of the updates made to the component state and instructs React to re-render the component and its children with the updated state

Q.6 Explain the concept of React fragments and when you should use them.

React Fragment is a feature in React that allows you to return multiple elements from a React component by allowing you to group a list of children without adding extra nodes to the DOM. To return multiple elements from a React component, you'll need to wrap the element in a root element

Q.7 How do you handle conditional rendering in React?

In React, conditional rendering is the process of displaying different content based on certain conditions or states. It allows you to create dynamic user interfaces that can adapt to changes in data and user interactions.

In this process, you can use conditional statements to decide what content should be rendered.

There is more than one way to do conditional rendering in React:

if

ternary operator

logical && operator

switch case operator

Conditional Rendering with enums