CA 4 Answers

1. Explain using code examples what is meant by props and state in React JS?

Props are arguments that get passed from a component to other components (like function parameters). A state stores information about a component. It is managed within the component and cannot be accessed by child components. In React data flow goes from parent to child, not the other way around.

Props example:

class Parent extends Component {

render() {

return (

<Child value="I am a prop" />

);

}

}

const Child = (props) => {

return <h1>{props.value}</h1>;

};

Here the Parent is passing the prop "I am a prop" to the Child. The Child can access the value of the prop by using props.value.

State example:

class Test extends React.Component {

constructor() {

this.state = {

id: 1,

status: "running"

};

}

render() {

return (

<div>

<p>{this.state.id}</p>

<p>{this.state.status}</p>

</div>

);

}

}

Here the state of the component is being initially set by the constructor. The values in the state are being rendered to the page.

To change the state of a component use setState:

this.setState({

status: "complete"

});

Here the state of the status has changed from running to complete. This may trigger something to happen on the page.

2. In functional programming, what does the term functor mean? Can you give an example in JavaScript?

A functor is a data structure that can store any data type. The map function can be used on a functor with the purpose of taking the values from it's container, modifying them, and then putting them back into the container. In JavaScript an array is a functor.

e.g.

let arr = [1, 2, 3];

arr = arr.map(val => val + 1); // arr will be [2, 3, 4]

Here the array arr is put through the map function to add 1 to each value in arr. It will change the values in arr to [2, 3, 4].

3. We have looked at three kinds of asynchronous programming mechanisms, namely callbacks, promises and streams. Mention one advantage and one disadvantage of each type.

**Callbacks**

Advantage - You can wait for the previous operation to be completed before executing the next operation.

Disadvantage - They do not scale well for asynchronous code that is in any way complex.

**Promises**

Advantage - They can handle multiple asynchronous operations easily while also providing better error handling than callbacks.

Disadvantage - They can't return multiple arguments.

**Streams**

Advantage - They are good at handling very large amounts of data of any type.

Disadvantage - Not suitable for applications with simple interactions with users or external services. Implementing streams for a system like this would be overkill.

4. With the aid of a diagram and example code, describe the Cascading Style Sheets (CSS) Box Model and show how it can be used to space DOM elements.

The CSS Box Model is a box that wraps around every HTML element. It consists of a margin, border, padding, and content.

Diagram, schematic

Description automatically generated

Content - The area of the page where text and images will be.

Padding - Generates space around the content.

Border - A border that surrounds the padding and content.

Margin - Generates space outside the border.

e.g.

<div>

<h1>Hello World</h1>

</div>

<style>

div {

width: 400px;

border: 15px solid red;

padding: 60px;

margin: 40px;

}

</style>

Shape, arrow

Description automatically generated

The Hello Word is the h1 inside the content. The content will be 400px wide. The content will be contained in a red 15px box. The box is the border. There will be 60px between the text and the border for padding and a 40px margin between the border and the edge of the page.

5. Detail how the browser loads and bootstraps a rich web application from an initial URL.

When a user accesses a URL, the browser goes to the Domain Name Servers (DNS) this is where the browser finds where the website is. The browser then connects to the server's IP address and the port assigned to the URL using a TCP/IP connection. These protocols tell the browser where on the server it can find the files to load the website. An application server or HTTP proxy will be listening at the server's IP address. It will accept the connection from the browser. There is now an open TCP/IP connection between the server and browser. The browser can now send HTTP requests to the server for the files it is looking for. The server will respond to the request by delivering the requested files.