WAD ASSIGNMENT 7

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8" />

<title>Book Example 11</title>

<script src="https://unpkg.com/react@16/umd/react.development.js"></script>

<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>

<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

<style>

#container {

padding: 50px;

background-color: #FFF;

}

</style>

</head>

<body>

<div id="container"></div>

<script type="text/babel">

var destination = document.querySelector("#container");

class Counter extends React.Component {

render() {

var textStyle = {

fontSize: 72,

fontFamily: "sans-serif",

color: "#333",

fontWeight: "bold"

};

console.log("render: Counter component");

return (

<div style={textStyle}>

{this.props.display}

</div>

);

}

}

class CounterParent extends React.Component {

constructor(props) {

super(props);

console.log("constructor: It's default time!");

this.state = {

count: 0

};

this.increase = this.increase.bind(this);

}

increase() {

this.setState({

count: this.state.count + 1

});

}

static getDerivedStateFromProps(props, state) {

console.log("getDerivedStateFromProps: You probably don't need this!")

return null;

}

componentDidUpdate(currentProps, currentState) {

console.log("componentDidUpdate: Component just updated!");

}

getSnapshotBeforeUpdate(prevProps, prevState) {

console.log("getSnapshotBeforeUpdate: Component is about to commit its changes!")

return null;

}

componentDidMount() {

console.log("componentDidMount: Component is inserted into the tree!");

}

componentWillUnmount() {

console.log("componentWillUnmount: Component is about to be removed from the DOM!");

}

shouldComponentUpdate(newProps, newState) {

console.log("shouldComponentUpdate: Should component update?");

if (newState.count < 5) {

console.log("shouldComponentUpdate: Component should update!");

return true;

} else {

ReactDOM.unmountComponentAtNode(destination);

console.log("shouldComponentUpdate: Component should not update!");

return false;

}

}

render() {

var backgroundStyle = {

padding: 50,

border: "#333 2px dotted",

width: 250,

height: 100,

borderRadius: 10,

textAlign: "center"

};

console.log("render: CounterParent component");

return (

<div style={backgroundStyle}>

<Counter display={this.state.count} />

<button onClick={this.increase}>

+

</button>

</div>

);

}

};

ReactDOM.render(

<div>

<CounterParent />

</div>,

destination

);

</script>

</body>

</html>

A screenshot of a computer

Description automatically generated