React Class Components

Introduction to Class Components in React

SoftUni Team
Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Have a Question?

Software University

sli.do

#react

Table of Contents

Class Components Definition

Features of Class Components

- Class components are **ES6** classes that extend **React.Component**, allowing for object-oriented features like inheritance
- They must implement a **render** method which returns **JSX**, defining what the UI should look like
- Class components have a built-in **state** object to manage local state, mutable via **this.setState()**

Class Component

- To define a **React component class**, you need to extend **React.Component**

```
class Person extends React.Component {
  render() {
    return <h1>My name is {this.props.name}</h1>
  }
}
```

  - The only method you must define is called **render()**

Lifecycle Methods in Class Components

Component Lifecycle

- A component has "**lifecycle methods**" that can be overridden to run code at times in the process
- A component has **3 lifecycle** phases
  - **Mounting**
  - **Updating**
  - **Unmounting**

Lifecycle Methods in Class Components

Software University

- Lifecycle methods in React class components provide hooks into different phases of a component's existence, from its initial **mounting** to **updates** and eventual **unmounting**
- These **methods** allow developers to execute code at specific points, making it easier to manage side effects, optimize performance, and handle cleanup tasks

Mounting

- **Mounting**
  - **componentDidMount()** Invoked immediately after a component is mounted, ideal for initializing network requests or setting up subscriptions

```
componentDidMount() {
 console.log('Component did mount');
}
```

Updating

- **Updating**
  - **componentDidUpdate()** Called after the component updates, suitable for performing DOM operations or fetching new data based on updated props or state

```
componentDidUpdate(prevProps, prevState) {
 if (prevState.count !== this.state.count) {
console.log('Component did update');
 }
}
```

Unmounting

- **Unmounting**
  - **componentWillUnmount()** Executed just before the component is unmounted and destroyed, perfect for cleanup tasks such as removing event listeners or canceling network requests

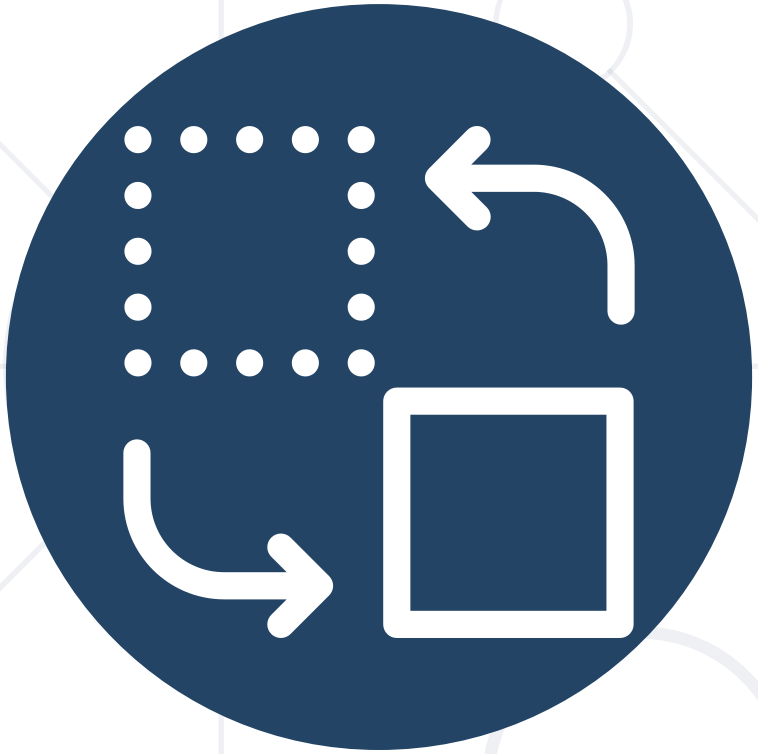
```
componentWillUnmount() {
  console.log('Component will unmount');
}
```

Component Props and State

Overview

Component Props

- **Props** is short for **properties**
  - Are received from above (parent)
  - **Immutable** as far as the component receiving them is concerned
- A component **cannot change** its own props, but it is responsible for putting together the props of its child components
- In **class components**, props are accessed via **this.props**, allowing the component to utilize these values within its **render** method and other class methods.



State and Class Components

Component State

State and Class Components

- In React, **class components** traditionally manage their own state using the **this.state** property
- To add state to a **class component**, you define a constructor method and set **this.state** to an object containing your state variables
- Unlike function components before hooks, **class components** have always been able to manage their own state using **this.state**

Updating State

- **State** is updated using the **this.setState()** method provided by React. It accepts an object that merges with the current state
- React automatically preserves the state between re-renders for class components.
- When **initializing** state in a class component, you define it in the constructor

State Initialization

- State Initialization

```
constructor(props) {
  super(props);
  this.state = {
   count: 0,
   // other state variables };
}
```

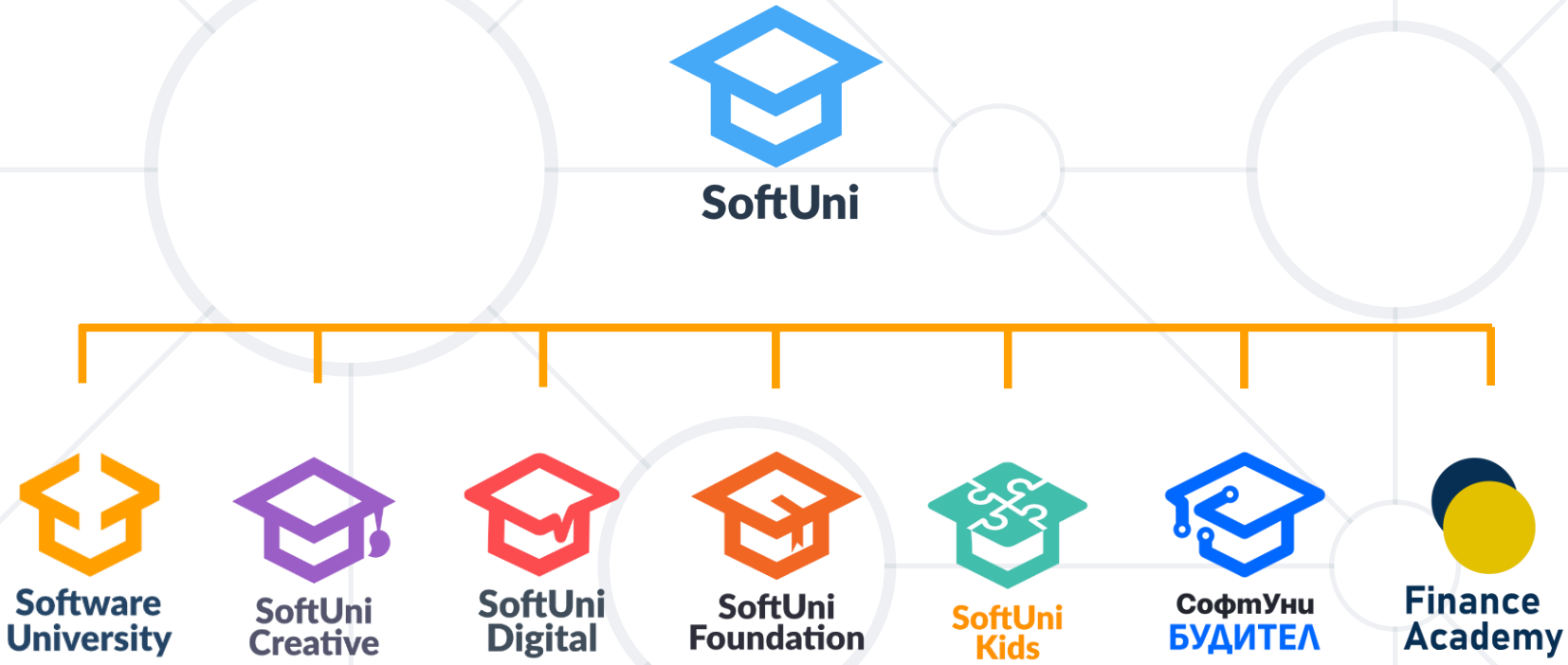
- To update state in a class component, you use **this.setState()**

```
this.setState({ count: this.state.count + 1 });
```

Summary

- Class components are JavaScript ES6 classes that extend React.Component
- Lifecycle methods like componentDidMount, componentDidUpdate, and componentWillUnmount allow you to run code at specific points in a component's life
- State in class components is initialized in the constructor with this.state

Questions?



SoftUni Diamond Partners

Software University

SmartIT

Coca-Cola HBC Bulgaria

SUPER HOSTING .BG

createX

Postbank

Решения за твоето утре

DRAFT KINGS

VIVACOM

AMBITIONED

INDEAVR

Serving the high achievers

SOFTWARE GROUP

PHAR VISION

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg
- © Software University – https://softuni.bg