Indhold

[Andre tutorials og vejledninger 2](#_Toc164342285)

[React's officielle dokumentation: 2](#_Toc164342286)

[freeCodeCamp: 2](#_Toc164342287)

[React Router dokumentation 2](#_Toc164342288)

[Styled Components dokumentation 2](#_Toc164342289)

[Jest dokumentation 2](#_Toc164342290)

[React Testing Library dokumentation 2](#_Toc164342291)

[Introduktion til React.js 3](#_Toc164342292)

[Opgave 3](#_Toc164342293)

[Besvarelse 4](#_Toc164342294)

[Component Lifecycle 5](#_Toc164342295)

[Opgave 5](#_Toc164342296)

[Besvarelse 6](#_Toc164342297)

[Tilstandsstyring 6](#_Toc164342298)

[Opgave 6](#_Toc164342299)

[Besvarelse 8](#_Toc164342300)

[React Router 9](#_Toc164342301)

[Opgave 9](#_Toc164342302)

[Besvarelse 10](#_Toc164342303)

[Form Håndtering 12](#_Toc164342304)

[Opgave 12](#_Toc164342305)

[Besvarelse 13](#_Toc164342306)

[Styling i React 14](#_Toc164342307)

[Opgave 14](#_Toc164342308)

[Besvarelse 15](#_Toc164342309)

[Tilpasning af React-komponenter 16](#_Toc164342310)

[Opgave 16](#_Toc164342311)

[Besvarelse 17](#_Toc164342312)

[Testning i React 18](#_Toc164342313)

[Opgave 18](#_Toc164342314)

[Besvarelse 19](#_Toc164342315)

Her er otte React.js-opgaver med udførlige kodeeksempler:

Udover Youtube videoen BRO-Code og W3CSchools kan i finde understøttelse til opgaverne nedenfor

Der er flere gode ressourcer online, hvor du kan finde tutorials og vejledninger til at lære React.js og løse ovenstående opgaver. Her er nogle af de mest populære:

# Andre tutorials og vejledninger

React's officielle dokumentation: React's officielle dokumentation er en fantastisk ressource, der dækker alt fra grundlæggende til avancerede emner. Du kan finde tutorials, guides og en omfattende API-reference der. Link: React Docs

freeCodeCamp: freeCodeCamp tilbyder gratis undervisningsmaterialer og tutorials, herunder en komplet guide til at lære React.js. De har også en aktiv fællesskabsforum, hvor du kan få hjælp og støtte. Link: freeCodeCamp React Guide

React Router dokumentation: Hvis du vil lære mere om React Router, er det værd at tjekke deres officielle dokumentation, der har omfattende guides og eksempler til at hjælpe dig med at implementere routing i din React-applikation. Link: React Router Docs

Styled Components dokumentation: For at lære mere om Styled Components, kan du tjekke deres officielle dokumentation, der tilbyder en grundig gennemgang af bibliotekets funktioner og hvordan man bruger det til at style React-komponenter. Link: Styled Components Docs

Jest dokumentation: Jest er et populært testrammeværktøj til JavaScript, herunder React-komponenttestning. Deres dokumentation giver omfattende vejledninger til at komme i gang med at skrive tests til dine React-komponenter. Link: Jest Docs

React Testing Library dokumentation: React Testing Library er et værktøj, der gør det nemt at teste dine React-komponenter, mens de opretholder en brugercentreret tilgang. Deres dokumentation indeholder mange nyttige eksempler og ressourcer til at hjælpe dig med at skrive effektive tests. Link: React Testing Library Docs

# Introduktion til React.js

## Opgave

Lav en simpel React-komponent kaldet `HelloWorld`, som viser en hilsen til brugeren.

## Besvarelse

```jsx

import React from 'react';

function HelloWorld() {

return <h1>Hello, World!</h1>;

}

export default HelloWorld;

```

# Component Lifecycle

## Opgave

Lav en komponent kaldet `Timer`, der viser en tæller, der starter fra 0 og opdateres hvert sekund.

## Besvarelse

```jsx

import React, { Component } from 'react';

class Timer extends Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

componentDidMount() {

this.interval = setInterval(() => {

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

}, 1000);

}

componentWillUnmount() {

clearInterval(this.interval);

}

render() {

return <h1>Count: {this.state.count}</h1>;

}

}

export default Timer;

```

# Tilstandsstyring

## Opgave

Lav en simpel tæller-applikation ved hjælp af `useState` hook'en.

## Besvarelse

```jsx

import React, { useState } from 'react';

function Counter() {

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

return (

<div>

<h1>Count: {count}</h1>

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

</div>

);

}

export default Counter;

```

# React Router

## Opgave

Lav en applikation med to sider, hvor brugeren kan navigere mellem siderne ved hjælp af React Router.

## Besvarelse

```jsx

// App.js

import React from 'react';

import { BrowserRouter as Router, Route, Link } from 'react-router-dom';

import Home from './Home';

import About from './About';

function App() {

return (

<Router>

<div>

<nav>

<ul>

<li><Link to="/">Home</Link></li>

<li><Link to="/about">About</Link></li>

</ul>

</nav>

<Route path="/" exact component={Home} />

<Route path="/about" component={About} />

</div>

</Router>

);

}

export default App;

// Home.js

import React from 'react';

function Home() {

return <h2>Home</h2>;

}

export default Home;

// About.js

import React from 'react';

function About() {

return <h2>About</h2>;

}

export default About;

```

# Form Håndtering

## Opgave

Lav en simpel formular med inputfelter til navn og e-mail, og vis indtastede data i en bekræftelsesbesked.

## Besvarelse

```jsx

import React, { useState } from 'react';

function Form() {

const [formData, setFormData] = useState({ name: '', email: '' });

const handleChange = (e) => {

setFormData({ ...formData, [e.target.name]: e.target.value });

};

const handleSubmit = (e) => {

e.preventDefault();

alert(`Name: ${formData.name}, Email: ${formData.email}`);

};

return (

<form onSubmit={handleSubmit}>

<input type="text" name="name" value={formData.name} onChange={handleChange} placeholder="Name" />

<input type="email" name="email" value={formData.email} onChange={handleChange} placeholder="Email" />

<button type="submit">Submit</button>

</form>

);

}

export default Form;

```

# Styling i React

## Opgave

Lav en komponent, der bruger Styled Components til at style en knap med en bestemt farve.

## Besvarelse

```jsx

import React from 'react';

import styled from 'styled-components';

const StyledButton = styled.button`

background-color: blue;

color: white;

padding: 10px 20px;

border: none;

border-radius: 5px;

`;

function StyledButtonComponent() {

return <StyledButton>Click me</StyledButton>;

}

export default StyledButtonComponent;

```

# Tilpasning af React-komponenter

## Opgave

Lav en genanvendelig komponent kaldet `Alert`, der tager en prop (`message`) og viser den i en beskedboks.

## Besvarelse

```jsx

import React from 'react';

function Alert({ message }) {

return <div className="alert">{message}</div>;

}

export default Alert;

```

# Testning i React

## Opgave

Lav en test til en simpel komponent, der tager et nummer som prop og viser det i en overskrift.

## Besvarelse

```jsx

// NumberComponent.js

import React from 'react';

function NumberComponent({ number }) {

return <h1>{number}</h1>;

}

export default NumberComponent;

```

```jsx

// NumberComponent.test.js

import React from 'react';

import { render } from '@testing-library/react';

import NumberComponent from './NumberComponent';

test('renders number correctly', () => {

const { getByText } = render(<NumberComponent number={42} />);

const numberElement = getByText(/42/i);

expect(numberElement).toBeInTheDocument();

});

```