

Компоненты высшего порядка

Цель работы

Получить навыки работы с компонентами высшего порядка.

Задания для выполнения

1. Ознакомьтесь со статьей <https://habr.com/ru/company/ruvds/blog/428572/>
2. Создайте компоненты по типу `InfoWithHover` и реализуйте компонент высшего порядка, который будет получать компоненты и возвращать компоненты со свойством `Hover`.
3. Релизуйте функционал для `Hover`.
4. Загрузить созданную страницу на GitHub в репозиторий `tera`, используя формат в названии `Фамилия (латинскими буквами)_15`.

Методические указания

Нам надо, чтобы наш компонент высшего порядка (назовём его `withHover`) мог бы инкапсулировать код обработки событий мыши, а затем передавал бы свойство `hovering` компонентам, которые он рендерит. Это позволит нам предотвратить дублирование соответствующего кода, разместив его в компоненте `withHover`.

1. Создадим компонент `Info` – это может быть иконка, картинка или элемент управления, например кнопка.
2. Создадим компонент `InfoWithHover`, который будет реализовывать функцию `withHover`.
3. Создадим функцию `Tooltip`, реализующую функционал `Hover` (Это может быть переход на другую страницу, показ изображения, осуществление анимации, запуск другого App).

```
import React, { StrictMode } from 'react';
import { createRoot } from 'react-dom/client';
```

```

import App from './App';

const rootElement = document.getElementById('root');
const root = createRoot(rootElement);
const InfoWithHover = withHover(Info)

function withHover(Component) {
  return class WithHover extends React.Component {
    state = { hovering: false }
    mouseOver = () => this.setState({ hovering: true })
    mouseOut = () => this.setState({ hovering: false })
    render() {
      return (
        <div onMouseOver={this.mouseOver} onMouseOut={this.mouseOut}>
          <Component hovering={this.state.hovering} />
        </div>
      );
    }
  }
}

function Info ({ hovering, height }) {
  return (
    <>
      {hovering === true
        ? <Tooltip id={this.props.id} />
        : null}
      <svg
        className="Icon-svg Icon--hoverable-svg"
        height={height}
        viewBox="0 0 16 16" width="16">
        <path d="M9 8a1 1 0 0 0-1-1H5.5a1 1 0 1 0 0 2H7v4a1 1 0 0 0 2 0zM4 0h8a4 4 0 0 1 4 4v8a4 4 0 0 1-4 4H4a4 4 0 0 1-4 4V4a4 4 0 0 1 4 4zm4 5.5a1.5 1.5 0 1 0 0 3 1.5 1.5 0 0 0 0 3z" />
        </svg>
      </>
    )
  }
}

function Tooltip ({ q }) {
  return (
    <>
      </>
    )
  }
}

root.render(
  <StrictMode>
    <App />
    <InfoWithHover />
  </StrictMode>
);

```

Вопросы для закрепления:

1. Чем компонент высшего порядка отличается от обычного компонента?
2. Может компонент высшего порядка принимать компонент, а возвращать свойства?

Дополнительные задания:

1. Реализуйте поиск на сайте с компонентами высшего порядка.

Полезные ссылки:

<https://habr.com/ru/company/ruvds/blog/428572/>

<https://stackblitz.com/edit/react-qwz9tu?file=src%2FApp.js,src%2Findex.js>