**Використання Render Props**

[**https://reactjs.org/docs/render-props.html**](https://reactjs.org/docs/render-props.html)

При описі компонентів ми можемо використати властивість-атрибут render (або якусь іншу властивість), якій передаємо функцію, що повертає компонент (для цього компонента не описуємо класу з методом render) . Цю функцію потім ми можемо використати у власній функції render, для генерування того компонента, який передано за допомогою функції.

|  |  |
| --- | --- |
|  |  |
| <компонент **render**={    функція, яка генерує компонент  )}/> | class Cat extends React.Component {  render() {  const mouse = this.props.mouse;  return (  <img src="/cat.jpg" style={{ position: 'absolute', left: mouse.x, top: mouse.y }} />  );  }  }  class Mouse extends React.Component {  constructor(props) {  super(props);  this.handleMouseMove = this.handleMouseMove.bind(this);  this.state = { x: 0, y: 0 };  }  handleMouseMove(event) {  this.setState({  x: event.clientX,  y: event.clientY  });  }  render() {  return (  <div style={{ height: '100%' }} onMouseMove={this.handleMouseMove}>  {/\*  Instead of providing a static representation of what <Mouse> renders,  use the `render` prop to dynamically determine what to render.  \*/}  {**this.props.render(this.state)**} **//Генеруємо компонент, який передано через властивість render**  </div>  );  }  }  class MouseTracker extends React.Component {  render() {  return (  <div>  <h1>Move the mouse around!</h1>  <Mouse **render={mouse => ( //Задаємо функцію, яка генерує компонент**  **<Cat mouse={mouse} />**  )}/>  </div>  );  }  } |