

Props, State и события

Цель работы

Познакомить с коллекциями значений, которые ассоциированы с компонентом. Эти значения позволяют создавать динамические компоненты, которые не зависят от жестко закодированных статических данных, а также учитывать состояние определенное внутри компонента и доступно только из компонента.

Задания для выполнения

1. Выведите свои ФИО, возраст в одной переменной с помощью props.
2. В конструкторе созданного класса установите объект state. Создайте кнопку, которая будет менять состояние при нажатии с помощью компонента React. При определении конструктора компонента в нем должен вызываться конструктор базового класса, в который передается объект props.
3. Загрузить созданную страницу на GitHub в репозиторий Student, используя формат в названии Фамилия (латинскими буквами)_6.

Методические указания

1. Для использования props можно воспользоваться синтаксисом на выбор:

Функциональный подход

```
function Hello(props) {  
  return <div>  
    <p>Имя: {props.name}</p>  
    <p>Возраст: {props.age}</p>  
  </div>;  
}
```

Использование классов ES6

```
class Hello extends React.Component {  
  render() {  
    return <div>  
      <p>Имя: {this.props.name}</p>  
      <p>Возраст: {this.props.age}</p>  
    </div>;  
  }  
}
```

С использованием стрелочной функции:

```
const Hello = (props) => {
```

```

const {name, age} = props;
return(<div>
  <p>Имя: {name}</p>
  <p>Возраст: {age}</p>
</div>);
}

```

А в конце использовать:

```

ReactDOM.render(
  <Hello name="Алевтина" age="33" />,
  document.getElementById("app")
)

```

2. Для обновления состояния вызывается функция setState():

```

this.setState({welcome: "Привет React"});

```

Вначале необходимо задать стиль для состояний:

```

button{
  width: 100px;
  height:30px;
  border-radius: 4px;
  margin:50px;
}
.on{
  color:#666;
  background-color: #ccc;
}
.off{
  color:#888;
  background-color: white;
}

```

А затем создать сам компонент изменяющий состояния:

```

class ClickButton extends React.Component {

  constructor(props) {
    super(props);
    this.state = {class: "off", label: "Нажми"};

    this.press = this.press.bind(this);
  }
  press(){
    let className = (this.state.class==="off")?"on":"off";
    this.setState({class: className});
  }
  render() {
    return <button onClick={this.press}
className={this.state.class}>{this.state.label}</button>;
  }
}

ReactDOM.render(
  <ClickButton />,
  document.getElementById("app")
)

```

И вызвать его в html-документе.

Контрольные вопросы

1. В чем отличие Props от State?
2. В чем разница между обработкой событий в React и элементов DOM с помощью обычного JavaScript

Дополнительные задания

1. Добавьте другие контролы к кнопке на страницу html.