**Вопросы 💎**

1. **Кратко опишите основные отличия state от props.**

props – это данные, переданные в компонент, их нельзя изменять, а state описывает внутреннее состояние компонента.

props передаётся в компонент (служат как параметры функции), в то время как state находится внутри компонента (по аналогии с переменными, которые объявлены внутри функции).

1. **Какую ошибку я допустила в следующем коде и как ее исправить:**

handleChange = () => {

const chechedArr = this.state.arr; // [1,2,3]

chechedArr.push(5); // [1,2,3,5]

this.setState({

arr: chechedArr,

});

};

Не уверена, что это правильно, но может надо вынести const chechedArr = this.state.arr; за пределы функции, изменяющей состояние.

1. **Можно ли повесить на один элемент несколько обработчиков событий?**

Да, можно.

1. **Каким образом можно изменить state?**

Нужно на компонент повесить обработчик события, который будет менять изначальный state (внутреннее состояние компонента). Для установки нового состояния предусмотрена функция setState (в классовом компоненте). Именно её вызов приводит к тому, что компонент перерисуется.

1. **Попробуйте предсказать какие сообщения будут в консоли и объяснить результат:**

handleChange = () => {

console.log(this.state.checked) //#1 false

this.setState({

checked: !this.state.checked,

});

console.log(this.state.checked) //#2 false

};

Будет false, т.к. React не обновляет `this.state.checked`, пока компонент не отрендерится снова. Получается, что второй console.log обращается к текущему значению `this.state.checked`, а это в первый раз false, поэтому в строке 2 тоже будет false.

Потом, когда мы кликнули на чекбокс, и состояние изменилось на true, в консоль выведется true и true в обоих случаях.

1. **Какими способами можно задать функцию handleChange и какой из них является самым правильным?**

Как обычная функция:

handleClick = () => {

const text = 'Hello world!';

this.setState({

text : text ,

});

};

Но с таким определением придется постоянно писать подобный код: onClick={this.handleClick.bind(this)}

Правильный способ определения - стрелочная функция.

handleClick = () => {

const text = 'Hello world!';

this.setState({ text });

};

Или

handleClick = () => {

const text = 'Hello world!';

this.setState({

text : text ,

});

};

1. **Чем отличаются классовые и функциональные компоненты? Какие из них предпочтительнее в 2021 году?**

Основное отличие функциональных компонентов от компонентов, основанных на классах, это то, что функциональные компоненты захватывают отрендеренные значения.

Пример из теории с кнопкой, по клику на которую с помощью функции setTimeout происходит задержка в 3 секунды, а потом выводится сообщение с подтверждением выполненной операции.

При клике на кнопку будут следующие особенности:

- При щелчке по кнопке, сформированной функциональным компонентом, при выбранном профиле `Dan` и при последующем переключении пока не прошли 3 секунды на профиль `Sophie`, в окне сообщения будет выведено `'Followed Dan'`.

- Если сделать то же самое с кнопкой, сформированной компонентом, основанном на классе, будет выведено `'Followed Sophie'`.

В функциональных компонентах происходит захват данных, поэтому изменение параметра не вызывает ошибку.

В 2021 предпочтительнее функциональные.

1. **Есть ли this в функциональных компонентах? Как можно получить к нему доступ?**

В функциональном компоненте недоступен this, поэтому мы не можем задать или считать состояние через this.state. Вместо этого вызываем хук useState напрямую изнутри нашего компонента. В useState мы передаем исходное состояние. Вызов useState вернёт пару значений: текущее состояние и функцию, обновляющую состояние.

1. **Можно ли использовать props и state одновременно?**

Можно, но нельзя менять сами props.

1. **Где можно задать state без использования команды this.setState?**

В функциональном компоненте.

1. **Может ли состояние классового компонента не быть объектом? А функционального?**

При использовании setState() в классовых компонентах переменная состояния всегда является объектом. В то время как переменная состояния в хуках может иметь любой тип, например, число, строку, логическое значение, объект или массив.

1. **Способны ли функциональные компоненты самостоятельно хранить состояние?**

мы можем сохранять состояние в функциональном компоненте с помощью хука useState.

1. **Как использовать хуки в классовых компонентах?**

В документации reactjs.org явно написано, что вы не можете использовать хуки внутри классового компонента, но вы можете комбинировать классы и функциональные компоненты с хуками в одном дереве.

1. **А как задать начальное состояние props, если они еще не были переданы?**

В React предусмотрен способ устанавливать значения пропсов по умолчанию — defaultProps.

Пример:

export default class Button extends React.Component {

static defaultProps = {

name: 'Hello!',

};

render() {

const {name, ...props} = this.props;

return (

<button {...props}>

{name}

</button>

);

}

}

Если мы не передадим в Button аттрибут name, то будет использовано значение по умолчанию "Hello!".