1. Кратко опишите основные отличия state от props. - *state и props оба являются JS-объектами, но у них разное предназначение. Если props представляет входные данные, которые передаются в компонент извне, то состояние хранит такие объекты, которые создаются в компоненте и полностью зависят от компонента.*
2. 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,

});

}; – *В данном коде ошибка заключается в изменении состояния компонента напрямую, используя ссылку на массив из state и метод push()*.

1. Можно ли повесить на один элемент несколько обработчиков событий? – *напрямую нет.*
2. Каким образом можно изменить state? – *нужно в обработчике событий передать функцию в которой описывается изменение state: this.setState({ checked: !this.state.checked, });*
3. Попробуйте предсказать какие сообщения будут в консоли и объяснить результат:

handleChange = () => {

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

this.setState({

checked: !this.state.checked, //true

});

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

}; – *будет false, так как изначально checked установлено как false.*

1. Какими способами можно задать функцию handleChange и какой из них является самым правильным? - *Стрелочная функция, обычная функция и связывание его с контекстом через bind, а также непосредственно в обработчике событий передача функции анонимной стрелочной функции. Самый правильный – стрелочная функция, так как с bind, если много элементов, то слишком много потребуется дописывать в коде, а передача анонимной функции может порождать при каждом вызове функции render новые обработчики (так как функции сравниваются по ссылкам, а не по содержимому).*
2. Чем отличаются классовые и функциональные компоненты? Какие из них предпочтительнее в 2021 году? – *предпочтительнее функциональные компоненты.*
3. Есть ли this в функциональных компонентах? Как можно получить к нему доступ? – t*his в функциональных компонентах нет, используется хук useState.*
4. Можно ли использовать props и state одновременно? – *можно, так как у них разное назначение, пропсы исподьзуются для константных значений, а state для изменяющихся.*
5. Где можно задать state без использования команды this.setState? – *в конструкторе и когда вытаскиваем свойство из объекта при помощи оператора расширения.*
6. Может ли состояние классового компонента не быть объектом? А функционального? – *в классовом компоненте state это объект, а в функциональном компоненте, состояние может, но не обязано, быть объектом.*
7. Способны ли функциональные компоненты самостоятельно хранить состояние? – *самостоятельно не могут, однако это можно сделать используя хуки.*
8. Как использовать хуки в классовых компонентах? – *хуки это замена state у классового компонента для выполнения аналогичных функции в функциональном компоненте, поэтому нет необходимости использовать его в классовых компонентах.*
9. А как задать начальное состояние props, если они еще не были переданы? – *использовать defaultProps.*