

# React: Hooks, State

# НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

# Повторим;)

■ Для чего используется props?

■ Как задать значения по умолчанию для prop?

■ В чем отличия prop children от остальных props?

# ЦЕЛЬ

Изучить варианты передачи функций в компонент, познакомиться с хуком `useState`

# ПЛАН ЗАНЯТИЯ

- 
- 1. Function in components
- 2. Hook useState

# Передача функций

**( )    =>    { }**

## Функция внутри onClick:

В этом варианте функция, отвечающая за действие, определяется непосредственно внутри onClick:

```
const ButtonComponent = () => {  
  
  return (  
    <div>  
      <button onClick={() => console.log('Click')}>Нажми  
меня</button>  
    </div>  
  );  
};
```

## Функция снаружи onClick:

В этом варианте функция, отвечающая за действие, определяется снаружи onClick:

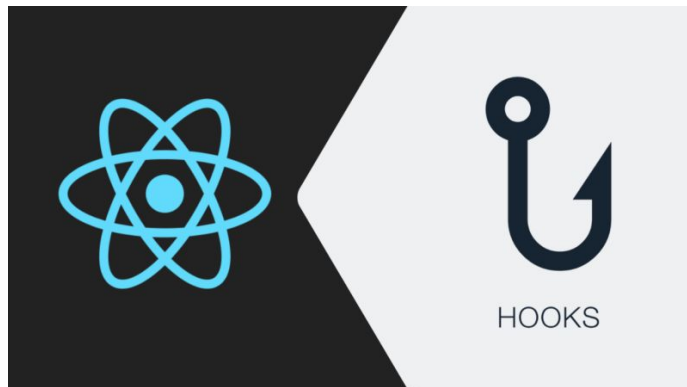
```
const ButtonComponent = () => {  
  
  const handleClick = () => {  
    console.log('Click');  
  };  
  
  return (  
    <div>  
      <button onClick={handleClick}>Нажми меня</button>  
    </div>  
  );  
};
```



# Передача аргументов

```
const ClickWithParameter = () => {  
  const handleClickWithParameter = (message) => {  
    console.log(`Кнопка была нажата. Сообщение: ${message}`);  
  };  
  
  return (  
    <button onClick={() => handleClickWithParameter('Привет, мир!')}>  
      Нажми меня (с параметром)  
    </button>  
  );  
};
```

# Хуки



**Хук — это специальная функция, которая позволяет «подцепиться» к возможностям React.**





# Правила хуков

1. **Используйте хуки только на верхнем уровне. Не вызывайте хуки внутри циклов, условных операторов или вложенных функций.**

1. **Вызывайте хуки только из React-функций**

**Не вызывайте хуки из обычных функций JavaScript. Вместо этого можно:**

-  **Вызывать хуки из функционального компонента React.**
-  **Вызывать хуки из пользовательского хука**

3. **Имя хука начинается со слова «use»**

# useState

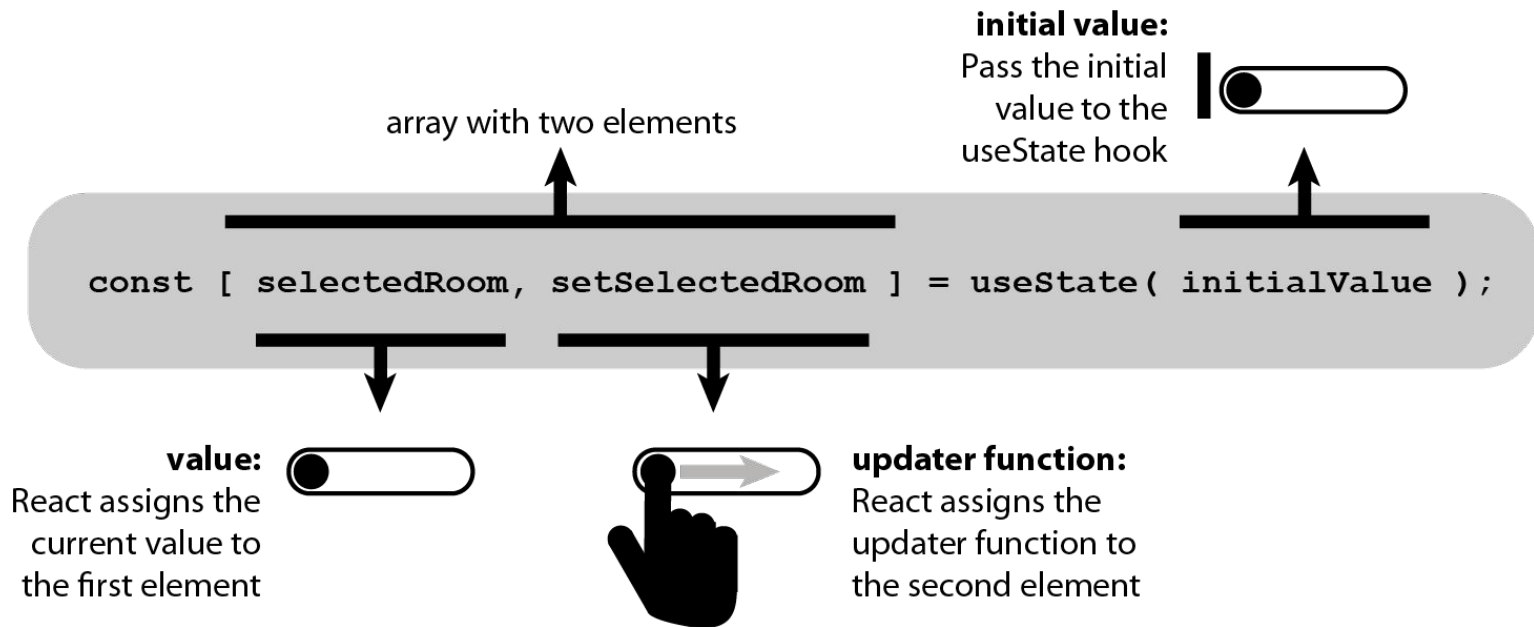


**useState** - это хук (hook) в библиотеке React, который позволяет функциональным компонентам иметь внутреннее состояние.

Состояние в React представляет собой данные, которые могут изменяться в течение времени, и когда состояние изменяется, React обновляет соответствующий компонент.



# Синтаксис useState



## Возможные значения initialValue

### 1. Числа

```
const [count, setCount] = useState(0);
```

### 1. Строки

```
const [name, setName] = useState('John');
```

### 1. Логические значения (boolean)

```
const [isVisible, setIsVisible] = useState(true);
```



## Возможные значения initialValue

### 4. Массивы

```
const [items, setItems] = useState(['apple', 'banana', 'orange']);
```

### 5. Объекты

```
const [person, setPerson] = useState({ name: 'John', age: 25 });
```

### 6. Функции

```
const [callback, setCallback] = useState(() => {  
  // какая-то логика  
  return 'результат';  
});
```

## Возможные значения initialValue

### 7. null или undefined:

```
const [data, setData] = useState(null);
```

Важно выбирать начальное значение в зависимости от типа данных, с которым вы работаете, и от того, что наиболее подходит для вашего конкретного случая использования.

Для работы с `useState` нужно сначала импортировать его из библиотеки

```
import React, { useState } from 'react';

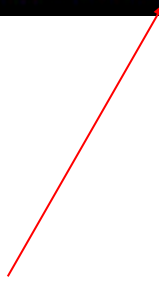
const MyComponent = () => {
  // Деструктуризация, где count - текущее состояние, setCount - функция для обновления
  const [count, setCount] = useState(0);

  // ...
};
```

## Использование значения из useState

Значение из useState может быть вставлена в любое место компонента в качестве переменной

```
<h2>Пример счетчика с функциональным обновлением</h2>  
<p>Текущее значение счетчика: {count}</p>
```



значение count из  
СОСТОЯНИЯ

## Обновление состояния

Второе значение, которое возвращает `useState` - это функция, которую предоставляет `useState` для обновления состояния. Она может принимать новое значение или функцию, которая возвращает новое значение на основе предыдущего состояния.

I

```
<button onClick={() => setCount(count + 1)}>Увеличить</button>
```

II

```
<button onClick={() => setCount(prevCount => prevCount + 1)}>Увеличить</button>
```



# **Ваша новая IT-профессия – Ваш новый уровень жизни**

Программирование с нуля в  
немецкой школе AIT TR GmbH