



Lecture 3 - Hooks

(useState, useEffect)



Что такое hook?

Hooks — это специальные функции, позволяющие «подключаться» к состоянию React и функциям жизненного цикла. Хуки позволяют функциональным компонентам управлять состоянием и побочными эффектами. Наиболее распространённые хуки: `useState`, `useEffect`, `useContext`, и т.д.



Что такое useState?

`useState` это `hook`, который позволяет добавить переменную состояния к компоненту.

Возвращает пару: текущее значение состояния и функцию для его обновления.

```
const [state, setState] = useState(initialState)
```



useState: объявление

`useState` следует вызывать на верхнем уровне компонента для объявления переменной состояния.

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

function MyComponent() {
  const [age, setAge] = useState(28);
  const [name, setName] = useState('Taylor');
  const [todos, setTodos] = useState(() => createTodos());
  // ...
}
```



useState: начальное значение

initialState: Значение, которое должно быть у состояния изначально. Это может быть значение любого типа, но для функций существует особое поведение. Этот аргумент игнорируется после первоначальной отрисовки.

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

function MyComponent() {
  const [age, setAge] = useState(28);
  const [name, setName] = useState('Taylor');
  const [todos, setTodos] = useState(() => createTodos());
  // ...
}
```



useState: set функция

Функция `set`, возвращаемая `useState`, позволяет обновить состояние до другого значения и запустить повторную визуализацию. Вы можете передать следующее состояние напрямую или передать функцию, которая вычисляет его на основе предыдущего:

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

function handleClick() {
  setName('Taylor');
  setAge(a => a + 1);
  // ...
}
```



Подводный камень

Вызов `set` функции **не** изменяет текущее состояние в уже выполняемом коде :

```
function handleClick() {  
  setName('Robin');  
  console.log(name); // Still "Taylor"!  
}
```

Это влияет только на то, что `useState` будет возвращено, начиная со *следующего* рендера.



Что такое useEffect?

`useEffect` это `hook`, который позволяет запускать побочные эффекты в функциональных компонентах.

```
useEffect(setup, dependencies?)
```




Что такое побочные эффекты??

Побочные эффекты — это действия, которые влияют на что-то за пределами компонента, например:

- Извлечение данных из API
- Подписка на события или веб-сокеты
- Обновление заголовка документа
- Запуск или очистка таймеров



useEffect: объявление

`useEffect` следует вызывать
на верхнем уровне
компонента для объявления
эффекта

```
import { useState, useEffect } from 'react';
import { createConnection } from './chat.js';

function ChatRoom({ roomId }) {
  const [serverUrl, setServerUrl] = useState('https://localhost:1234');

  useEffect(() => {
    const connection = createConnection(serverUrl, roomId);
    connection.connect();
    return () => {
      connection.disconnect();
    };
  }, [serverUrl, roomId]);
  // ...
}
```



useEffect: параметры

1. **setup:**

Функция с логикой вашего эффекта. При **mount** компонента React запускает функцию **setup**.

- При изменении зависимости: React выполняет очистку (если предусмотрено) со старыми значениями, а затем снова выполняет настройку
- При **unmount**: React выполняет очистку в последний раз.

```
import { useState, useEffect } from 'react';
import { createConnection } from './chat.js';

function ChatRoom({ roomId }) {
  const [serverUrl, setServerUrl] = useState('https://localhost:1234');

  useEffect(() => {
    const connection = createConnection(serverUrl, roomId);
    connection.connect();
    return () => {
      connection.disconnect();
    };
  }, [serverUrl, roomId]);
  // ...
}
```



useEffect: Parameters

1. **dependencies:**

Список значений (свойств, состояний или переменных), используемых в эффекте.

- Записывается в виде [dep1, dep2].
- React перезапускает эффект только при изменении зависимости.
- Пустой массив [] → выполняется один раз при mount.
- Без массива → выполняется после каждого рендеринга.

```
import { useState, useEffect } from 'react';
import { createConnection } from './chat.js';

function ChatRoom({ roomId }) {
  const [serverUrl, setServerUrl] = useState('https://localhost:1234');

  useEffect(() => {
    const connection = createConnection(serverUrl, roomId);
    connection.connect();
    return () => {
      connection.disconnect();
    };
  }, [serverUrl, roomId]);
  // ...
}
```



useEffect: возвращает

useState возвращает undefined.