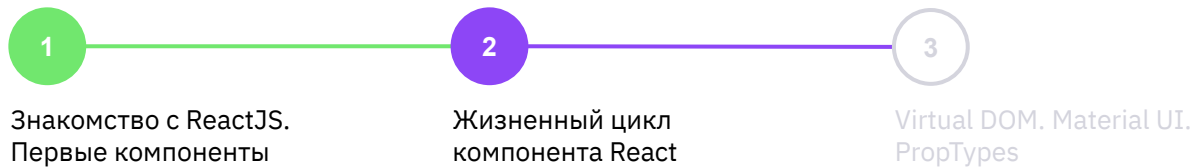


# Жизненный цикл компонента React

State, обновление компонента и хуки



## План курса





## Что будет на уроке сегодня



State



Жизненный цикл компонента



Хуки Реакт



Рендер массивов. Фрагмент



## State (состояние)

### Стейт (state)

Специальная переменная, сохраняющая присвоенное ей значение внутри компонента.

```
1  const [count, setCount] = useState(0);  
2  
3  // то же самое, что  
4  const countState = useState(0);  
5  const count = countState[0];  
6  const setCount = countState[1];  
7
```



## Обновление state

```
1 import React, { useState } from 'react';
2
3 export function Counter() {
4   const [count, setCount] = useState(0);
5
6   const updateCount = () => {
7     setCount(count + 1);
8   }
9
10  return (
11    <div>
12      <span className="counter">{count}</span>
13      <button className="counter-button" onClick={updateCount}>+1</button>
14    )
15  }
```



## Жизненный цикл

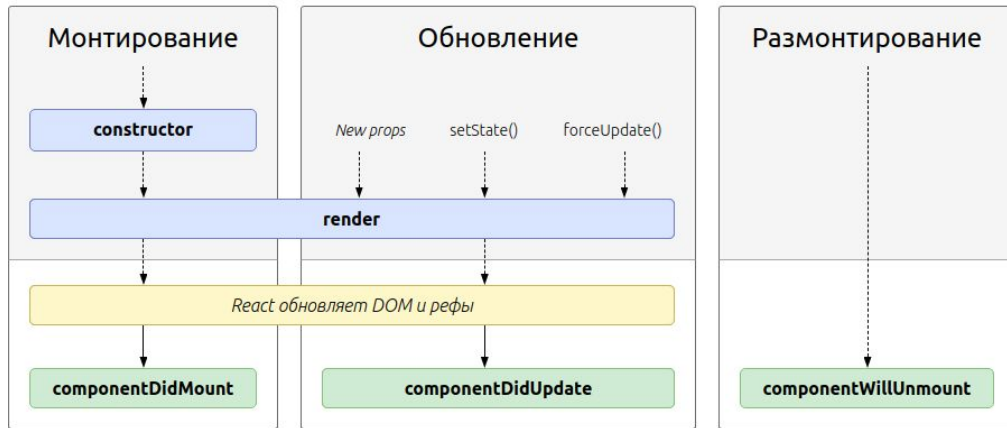
### “Этап Render”

Чистый и не имеет побочных эффектов.  
Может быть приостановлен, прерван и перезапущен самим React.

---

### “Этап Commit”

Может работать с DOM, выполнять  
побочные эффекты, назначать  
обновления.





## Хуки

**Хук (hook) в Реакт** – специальная функция, которая вызывается в теле функционального компонента и используется, в первую очередь, для сохранения значений переменных между обновлениями компонента



## Основные хуки

- `useState`
- `useEffect`
- `useCallback`
- `useRef`
- `useMemo`





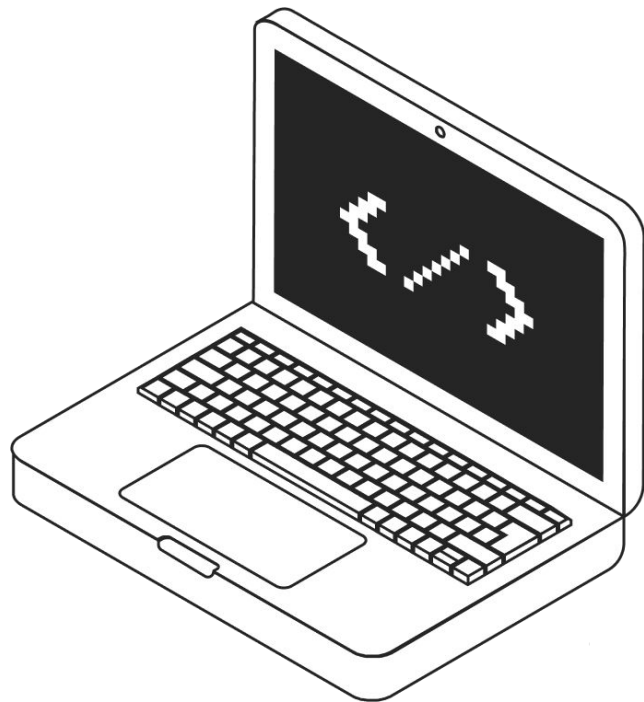
## Хуки – пример

```
1 const changeCount = useCallback(() => {  
2   setCount(1)  
3 }, []);  
4  
5 const elem = useMemo(  
6   () => props.hugeArray.find(el => el.show),  
7   [props.hugeArray]  
8 );  
9
```



## Правила хуков

- Вызывать только в функциональных компонентах или своих кастомных хуках
- Вызывать только на верхнем уровне (не в теле циклов или условий)





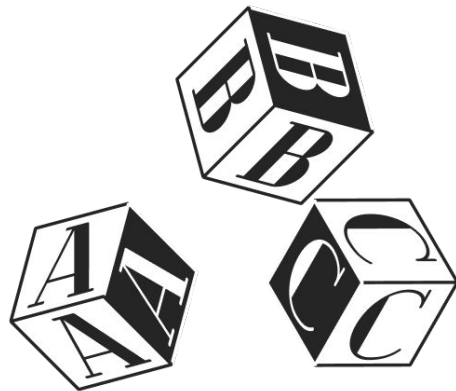
## Рендер массивов

```
1 function MessagesList() {  
2   const [messages, setMessages] = useState([  
3     "message 1",  
4     "message 2",  
5     "message 3",  
6   ]);  
7  
8   return messages.map((message) => <div>{message}</div>);  
9 }
```



## Фрагмент

Специальный компонент React, он служит для группировки нескольких элементов. При рендере в DOM он не будет добавлен как элемент





## Фрагмент

```
1    <>
2      <span>This is right!</span>
3      <div>Краткая запись фрагмента</div>
4    </>
5    // ...
6    <React.Fragment>
7      <span>This is right!</span>
8      <div>Полная запись фрагмента</div>
9    </React.Fragment>
```



**Спасибо за внимание**

