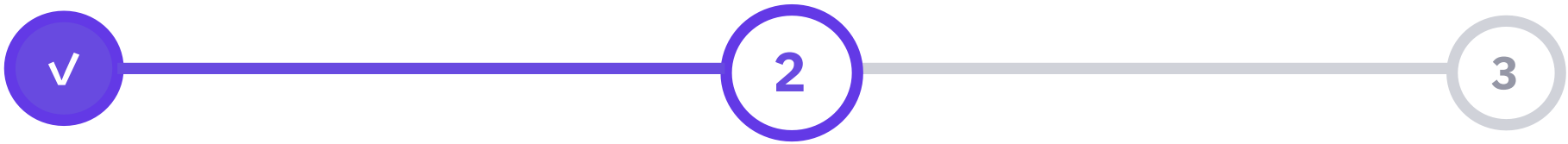
Жизненный цикл компонента React

State, обновление компонента и хуки



Урок 1
Знакомство с ReactJS. Первые компоненты

Урок 2
Жизненный цикл компонента React

Урок 3
Virtual DOM. Material UI. PropTypes

Что будет на уроке?

- **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

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

export function Counter() {
 const [count, setCount] = useState(0);

 const updateCount = () => {
   setCount(count + 1);
 }

 return (
   <div>
     <span className="counter">{count}</span>
     <button className="counter-button" onClick={updateCount}>+1</button>
 )
}
```

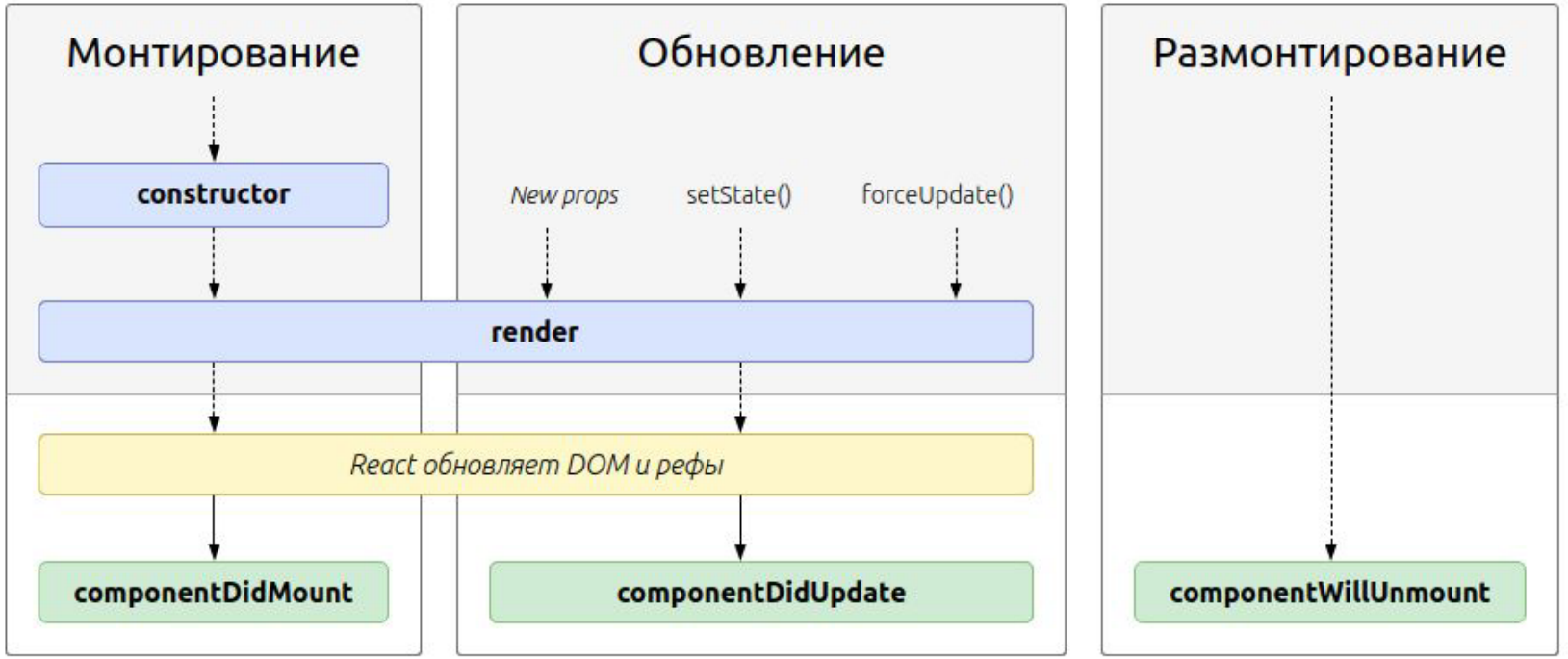
Жизненный цикл

“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

Фрагмент

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

Фрагмент

```
<>
  <span>This is right!</span>
  <div>Краткая запись фрагмента</div>
</>
// ...
<React.Fragment>
  <span>This is right!</span>
  <div>Полная запись фрагмента</div>
</React.Fragment>
```

Практическое задание

1. Добавить в компонент App поле стейта messageList. В нем хранить массив объектов - сообщений (объект должен содержать, как минимум, поля text и author). Начальное значение - пустой массив.
2. Рендерить список сообщений через map.
3. Добавить и стилизовать форму - поле для ввода текста и кнопка для отправки сообщения. При отправке сообщения обновление UI должно происходить за счет обновления стейта App.

Практическое задание

4. Добавить в App useEffect - на каждое отправленное пользователем сообщение должен отвечать робот (должно автоматически отправляться сообщение с фиксированным текстом) - для этого необходимо проверять автора последнего сообщения.
5. * Сделать задержку ответа робота - сообщение от него должно отправляться через некоторый промежуток времени после отправки сообщения пользователя (напр., 1.5 сек).

Заключительный слайд

Спасибо!

Задавайте вопросы