

План лекции

> Что такое React?

> Компоненты

> JSX

> Жизненный цикл

> Virtual DOM

Что такое React

React — это декларативная, эффективная и гибкая JavaScript библиотека для создания пользовательских интерфейсов. Она позволяет вам собирать сложный UI из маленьких изолированных кусочков кода, называемых «компонентами».

КОМПОНЕНТЫ

```
8
9
10 class Hello extends React.Component {
11     render() {
12         return <div> PARMA technologies group!</div>;
13     }
14 }
15
16
```

```
9
10
11 const Hello = () => {
12     return <div> PARMA technologies group!</div>;
13 };
14
15
```

JSX

```
8
9
10 class Hello extends React.Component {
11     render() {
12         return <div> PARMA technologies group!</div>;
13     }
14 }
15
16
```

JSX

JSX - это обертка над React.createElement

```
const virtualDom = <div className="parma"> PARMA  
technologies group</div>;
```

```
"use strict";
```

```
var virtualDom =  
/*#__PURE__*/React.createElement("div", {  
  className: "parma"  
}, " PARMA technologies group");
```

```
<div className="card">
  <div className="body">
    <h4 className="title">title</h4>
    <p className="text">my text</p>
    <a href="#" className="btn">Go</a>
  </div>
</div>
```

```
"use strict";
```

```
/*#__PURE__*/
React.createElement("div", {
  className: "card"
}, /*#__PURE__*/React.createElement("div", {
  className: "body"
}, /*#__PURE__*/React.createElement("h4", {
  className: "title"
}, "title"), /*#__PURE__*/React.createElement("p", {
  className: "text"
}, "my text"), /*#__PURE__*/React.createElement("a", {
  href: "#",
  className: "btn"
}, "Go"))));
```

СВОЙСТВО Props

```
class Componets extends React.Component {  
  render() {  
    return <div>{this.props.text}</div>  
  }  
}
```

```
const Componets = (props) => {  
  return <div>{props.text}</div>  
}
```

Обработка коллекций

```
class List extends React.Component {  
  render() {  
    const { list } = this.props;  
    return (  
      <ul>  
        {list.map((item) => (  
          <li>{item}</li>  
        ))}  
      </ul>  
    );  
  }  
}
```



```
class List extends React.Component {  
  renderList() {  
    // ...  
  }  
  
  render() {  
    return (  
      <ul>  
        {this.renderList()}  
      </ul>  
    );  
  }  
}
```

```
class List extends React.Component {  
  render() {  
    const { list } = this.props;  
    return (  
      <ul>  
        {list.map((item) => (  
          <li key={item.id}>{item}</li>  
        ))}  
      </ul>  
    );  
  }  
}
```

Children

```
class Welcome extends React.Component {  
  render() {  
    return <p>{this.props.children}</p>;  
  }  
}
```

State

```
import React from "react";

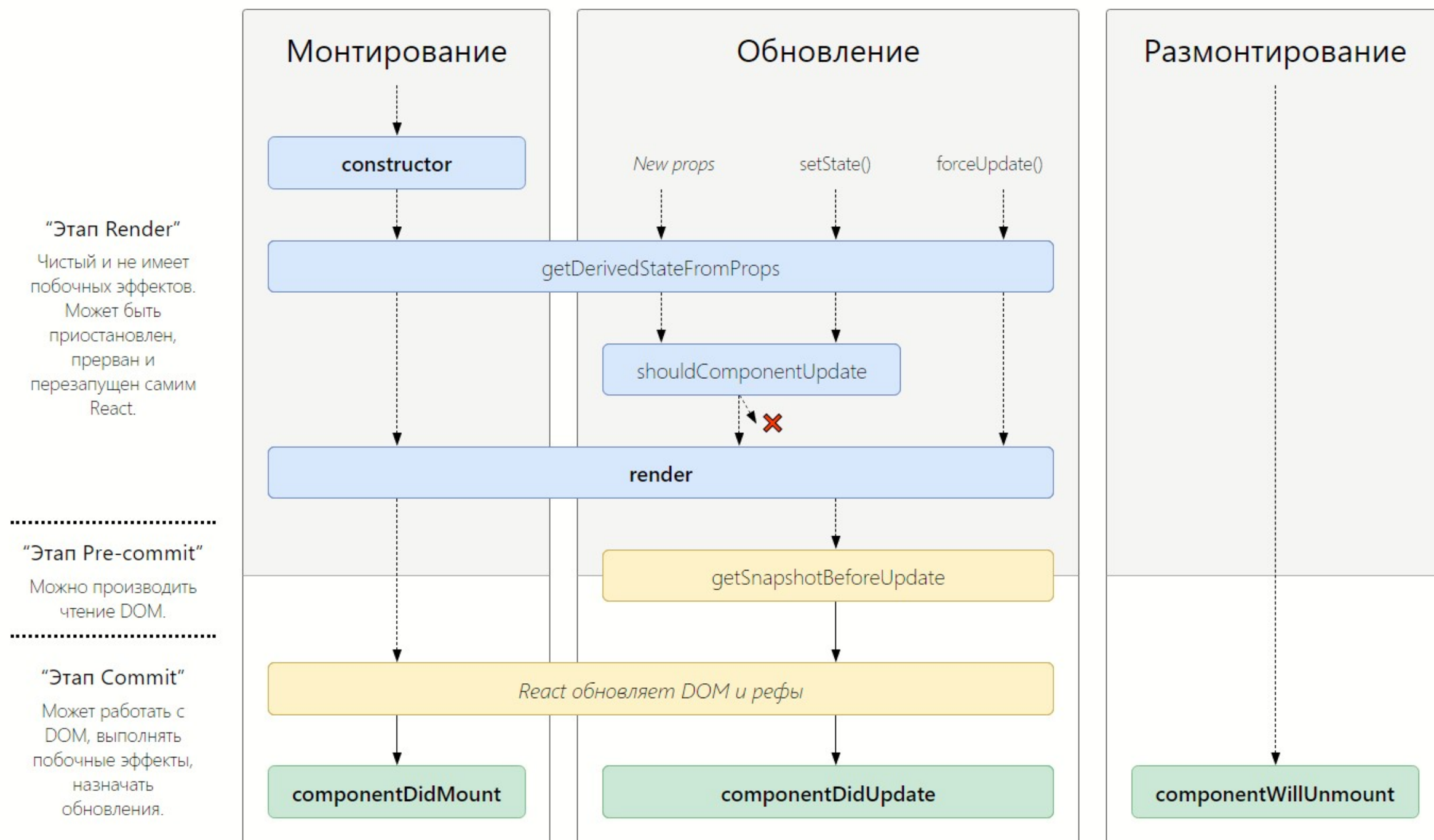
class Componets extends React.Component {
  constructor(props) {
    super(props); // всегда обязательно
    this.state = { count: 0 };
  }

  onHandlerClick = () => {
    const { count } = this.state;
    this.setState({ count: count + 1 });
  };

  render() {
    const { count } = this.state;
    return (
      <div>
        <h1>{count}</h1>
        <button onClick={this.onHandlerClick}>+1</button>
      </div>
    );
  }
}

export { Componets };
```

Жизненный цикл



Virtual DOM

Что такое виртуальный DOM?

Виртуальный DOM (VDOM) — это концепция программирования, в которой идеальное или «виртуальное» представление пользовательского интерфейса хранится в памяти и синхронизируется с «настоящим» DOM при помощи библиотеки, такой как ReactDOM. Этот процесс называется согласованием.

