React

Библиотека для создания пользовательских интерфейсов

Hello, world!

```
<div id="react-app"></div>

<script src="https://unpkg.com/react@16/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>

<script>
    let rootElement = React.createElement(
        'div',
        { style: { color: 'green' } },
        'Hello, world!'
    );

    ReactDOM.render(
        rootElement,
        document.getElementById('react-app')
    );
</script>
```

Сложные структуры элементов

```
let rootElement =
    React.createElement('div', {},
        React.createElement('h1', {}, "Contacts"),
        React.createElement('ul', {},
            React.createElement('li', {},
                React.createElement('h2', {}, "Bill Gates"),
                React.createElement('a', { href: 'mailto:bill@gates.com' }, 'bill@gates.com')
            ),
            React.createElement('li', {},
                React.createElement('h2', {}, "Steve Jobs"),
                React.createElement('a', { href: 'mailto:steve@jobs.com' }, 'steve@jobs.com')
            ),
            React.createElement('li', {},
                React.createElement('h2', {}, "Jeff Bezos")
            )
        )
    );

ReactDOM.render(rootElement, document.getElementById('react-app'));
```

Создание react элементов из данных

```
let contacts = [
    { name: 'Bill Gates', email: 'bill@gates.com' },
    { name: 'Steve Jobs', email: 'steve@jobs.com' },
    { name: 'Jeff Bezos' }
];

let contactElements = contacts.map(c => {
    return React.createElement('li', { key: c.name },
        React.createElement('h2', {}, c.name),
        c.email != undefined
            ? React.createElement('a', { href: c.email }, c.email)
            : null
    )
});

let rootElement =
    React.createElement('div', {},
        React.createElement('h1', {}, "Contacts"),
        React.createElement('ul', {}, contactElements)
    );
```

Компонент Email

```
class Email extends React.Component {
    render() {
        return React.createElement(
            'a',
            { href: this.props.value },
            this.props.value
        );
    }
}
```

Компонент Contact

```
class Contact extends React.Component {
    render() {
        let emailElement;

        if (this.props.email) {
            emailElement = React.createElement(Email, { value: this.props.email });
        }

        return React.createElement('li', {},
            React.createElement('h2', {}, this.props.name),
            emailElement
        );
    }
}
```

Компонент App

```
class App extends React.Component {
    render() {
        let contactElements = this.props.contacts.map(c =>
            React.createElement(Contact,
                {
                    key: c.name,
                    name: c.name,
                    email: c.email
                })
        );

        return React.createElement('div', {},
            React.createElement('h1', {}, "Contacts"),
            React.createElement('ul', {}, contactElements)
        );
    }
}
```

Приложение на компонентах

```
let contacts = [
    { name: 'Bill Gates', email: 'bill@gates.com' },
    { name: 'Steve Jobs', email: 'steve@jobs.com' },
    { name: 'Jeff Bezos' }
]

let rootElement = React.createElement(App, { contacts })

ReactDOM.render(rootElement, document.getElementById('react-app'));
```

Изменение данных и перерисовка

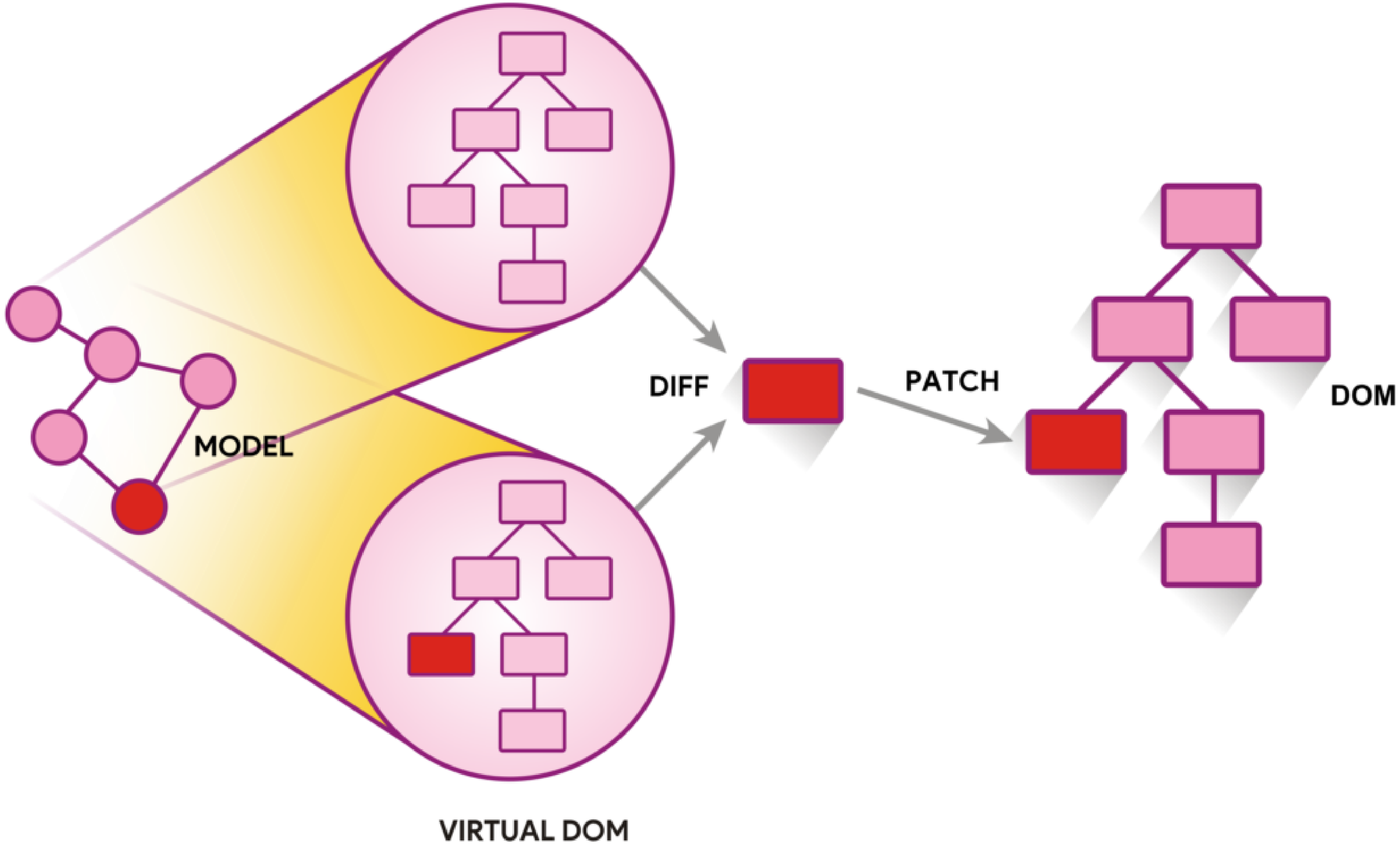
```
let contacts = [
    { name: 'Bill Gates', email: 'bill@gates.com' },
    { name: 'Steve Jobs', email: 'steve@jobs.com' },
    { name: 'Jeff Bezos' }
];

// first render
let rootElement = React.createElement(App, { contacts });
ReactDOM.render(rootElement, document.getElementById('react-app'));

// changing data
contacts[0].email = 'bill@mail.ru';
contacts.push({ name: 'Sergey Brin', email: 'sergey@brin.com' });

// render again
let nextRootElement = React.createElement(App, { contacts });
ReactDOM.render(nextRootElement, document.getElementById('react-app'));
```

Как работает React



JSX синтаксис

```
let element = React.createElement("div", {},
    React.createElement("h3", {}, "ToDo list"),
    React.createElement("ul", {},
        React.createElement("li", {}, "Learn JS"),
        React.createElement("li", {},
            "Learn ",
            React.createElement("a",
                    { href: "https://reactjs.org" },
                "React"
            )
        )
    )
);
```

```
let element = (
    <div>
        <h3>ToDo list</h3>
        <ul>
            <li>Learn JS</li>
            <li>
                Learn
                <a href="https://reactjs.org">
                    React
                </a>
            </li>
        </ul>
    </div>
);
```

Слева код на JS, справа – аналог на JSX.