# Способы описания компонент с реквизитами. Деструктурирование

### Цель работы

Познакомить со способами описания компонент в React, а также как передать реквизиты компоненту, как прочитать реквизиты из компонента, как указать значения по умолчанию для реквизитов, как передать некоторый JSX компоненту, как реквизиты меняются со временем.

### Задания для выполнения

1. Выполните тест: <https://www.schoolsw3.com/react/exercise.php?filename=exercise_es6_arrow1>
2. Подключите компилятор Babel к html-странице.
3. Определите один и тот же компонент тремя способами: функционально, используя классы ES6, с помощью стрелочной функции.
4. Ознакомьтесь с тем, как можно передавать атрибуты дочерним компонентам: <https://react.dev/learn/passing-props-to-a-component>

4. Создайте сайт со стилями Bootstrap для книжного магазина, в котором будут реализованы:

- вложенность, передача реквизитов компоненту;

- пересылка реквизитов с помощью синтаксиса распространения JSX (используйте синтаксис spread);

- в случае неизменимых реквизитов указано значение по умолчанию, подобное size = 100, которое используется для отсутствующих реквизитов и undefined;

- иерархия компонентов (есть родительские и дочерние),

- все повторяющиеся элементы реализованы через списки массивов с соответствующими ключами (лаба 3),

- стили реализовать по примеру (<https://bombora.ru/> или <https://www.mann-ivanov-ferber.ru/> ):







1. Загрузить созданную страницу на GitHub https://github.com/al-shatalova/Lab-4.git, используя формат в названии Фамилия, Группа (латинскими буквами)\_4.

### Методические указания

*1. Чтобы подключить Babel:*

В <head> укажите

<script src="<https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.25.0/babel.min.js>"></script>

А также элемент script, который содержит основной код приложения, должен иметь атрибут type="text/babel":

<script type="text/babel">

    ReactDOM.render(

        <h1>Hello React</h1>,

        document.getElementById("app")

    )

</script>

*2. Чтобы задать компонент функционально:*

function Hello() {

  return <h1>Привет</h1>;

}

*Чтобы задать компонент с помощью стрелочной функции:*

var Hello =() => {

    return (<h1>Привет</h1>);

}

*Чтобы задать компонент с помощью классов ES6:*

class Hello extends React.Component {

  render() {

    return <h1>Привет</h1>;

  }

}

*Также нужно задать метод render , который возвращает создаваемый элемент на JSX.*

ReactDOM.render(

 <Hello />,

 document.getElementById("container")

 )

3. Компонент с таймером:

class Timer extends React.Component {

 constructor(props) {

 super(props);

 this.state = { seconds: 0 };

 }

 tick() {

 this.setState(state => ({

 seconds: state.seconds + 1

 }));

 }

 componentDidMount() {

 this.interval = setInterval(() => this.tick(), 1000);

 }

 componentWillUnmount() {

 clearInterval(this.interval);

 }

 render() {

 return (

 <div>

 Seconds: {this.state.seconds}

 </div>

 );

 }

}

ReactDOM.render(

 <Timer />,

 document.getElementById('timer-example')

);

Примеры компонент представлены здесь:

<https://reactjs.org/>

### Контрольные вопросы

1. Зачем подключать Babel?
2. Что плохого в описании компонентов с использованием механизма классов?