

JSX в React

№ урока: 2 **Курс:** React Essential

Средства обучения: Браузер Chrome, редактор кода VS Code или любой другой.

Обзор, цель и назначение урока

Узнать, что такое JSX, научиться писать простые компоненты с помощью JSX. Рассмотреть применение условных операторов.

Изучив материал данного занятия, учащийся сможет

- Писать простые компоненты.
- Писать компоненты с помощью JSX.
- Использовать условные операторы в JSX.

Содержание урока

1. Что такое JSX и как он работает
2. Создание простого компонента
3. Передача параметров (props)
4. Передача контента
5. Работа с условными операторами

Резюме

- **JSX** — синтаксический сахар, расширение языка JavaScript. JSX напоминает язык шаблонов, наделённый силой JavaScript. JSX производит «элементы» React.
- **Элементы React** — это простые объекты, не отнимающие много ресурсов, а также мельчайшие кирпичики React-приложений. Элемент описывает то, что вы хотите увидеть на экране.
- После компиляции каждое **JSX-выражение** становится обычным вызовом JavaScript-функции, результат которого — объект JavaScript. Из этого следует, что JSX можно использовать внутри выражений **if** и **циклов for**, присваивать переменным, передавать функции в качестве аргумента и возвращать из функции.
- **Babel.JS** — это транспайлер, переписывающий код на ES6 (ES-2015) в код на предыдущем стандарте ES5.

Он состоит из двух частей:

- 1) Собственно транспайлер, который переписывает код.
- 2) Полифил, который добавляет методы `Array.from`, `String.prototype.repeat` и другие.

- Babel компилирует JSX в вызовы `React.createElement()`.
- React DOM использует стиль именования `camelCase` для свойств, вместо обычных имён HTML-атрибутов.
- **Компоненты** - позволяют разбить интерфейс на независимые части, про которые легко думать в отдельности. Их можно складывать вместе и использовать несколько раз.

- Компонент может быть классовым или функциональным.
- **Props** - React собирает все JSX-атрибуты и дочерние элементы в один объект и передаёт их нашему компоненту. Этот объект называется «пропс» (props).
- Пропсы можно только читать. Компонент никогда не должен что-то записывать в свои пропсы, вне зависимости от того, функциональный он или классовый.
- **Props.children** - также можно передать данные в середину компонента поместив их между открывающим и закрывающим тегом. Получить доступ к данным можно с помощью свойства children в объекте props.
- Вы можете внедрить любое выражение в JSX, заключив его в фигурные скобки. Это правило распространяется и на логический оператор && языка JavaScript.
- **Для рендеринга React-элемента в корневой узел DOM**, вызовите ReactDOM.render() с React-элементом и корневым DOM узлом в качестве аргументов.
- В редких случаях может потребоваться позволить компоненту спрятать себя, хотя он уже и отрендерен другим компонентом. Чтобы этого добиться, верните null вместо того, что обычно возвращается на рендеринг.

Закрепление материала

- Зачем нужно импортировать React в файле, где мы пишем JSX?
- Как встроить выражение Javascript в JSX?
- Во что превращается JSX-выражение после компиляции?
- Какой стиль именования свойств использует React-DOM?
- С какой буквы нужно писать название React компонентов?
- Можем ли мы внедрить любое выражение Javascript в JSX?

Дополнительное задание

Задание

Создать приложение с помощью Create React App, вывести на странице текст «hello world».

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Ознакомьтесь с разделом документации «Знакомство с JSX» и «Рендеринг элементов».

Задание 3

Создайте компонент, который будет принимать параметром строки и отображать их в разных блоках div.

Рекомендуемые ресурсы

<https://ru.reactjs.org/docs/components-and-props.html#function-and-class-components>

<https://ru.reactjs.org/docs/introducing-jsx.html#embedding-expressions-in-jsx>

<https://ru.reactjs.org/docs/react-dom.html#render>