

Создание списка заметок

№ урока: 7 **Курс:** React Essential

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

Обзор, цель и назначение урока

Узнать, как можно выбирать тип компоненты во время исполнения. Попрактиковаться применять знания props, state, lifecycle и другие. Создать список заметок с помощью React.

Изучив материал данного занятия, учащийся сможет

- Разбивать компоненты на более мелкие части.
- Выбирать тип компоненты во время исполнения.
- Создавать небольшие компоненты с применением state, lifecycle методов.

Содержание урока

1. Разбиение на компоненты
2. Верстка списка
3. Компонент Кнопка
4. Добавление и удаление элементов

Резюме

- **The «React way» to render a list:**
 - o Использовать Array.map вместо for loop;
 - o Присвоить каждому элементу списка уникальный атрибут «key»;
 - o Избегать использование индекса как ключа;
 - o Не создавать лишние элементы обертки если рендерите массив.
 - **Указание типа React-элемента:**
 - o Использование записи через точку. Вы можете ссылаться на React-компонент, используя запись через точку. Это удобно, если у вас есть модуль, который экспортирует много React-компонентов. Названия типов пользовательских компонентов должны начинаться с большой буквы.
 - o Выбор типа во время исполнения. В качестве типа React-элемента нельзя использовать выражение. Если вы хотите использовать выражение, чтобы указать тип элемента, присвойте его в переменную, начинающуюся с заглавной буквы.
 - **Атрибуты расширения** - Если у вас уже есть пропсы внутри объекта «props» и вы хотите передать их в JSX, вы можете использовать оператор расширения «...».
- Вы так же можете выбрать конкретные пропсы, которые ваш компонент будет использовать, передавая все остальные пропсы с помощью оператора расширения.
- **Функции как дочерние компоненты** - Обычно JavaScript-выражения, вставленные в JSX, будут приведены к строке, React-элементу или списку из всего этого. Тем

не менее, `props.children` работает так же, как и любой другой проп, поэтому в него можно передавать любые типы данных, а не только те, которые React знает как рендерить. К примеру, если у вас есть пользовательский компонент, можно было бы передать колбэк в «`props.children`».

Дочерние компоненты, передаваемые пользовательскому компоненту, могут быть чем угодно с тем условием, что компонент преобразует их во что-то, что React сможет понять и отрендерить.

- **Атрибут `key`** - единственная вещь, которую React использует для идентификации DOM-элементов. Что произойдёт, если Вы поместите элемент в список или удалите что-то посередине? Если `key` точно такой же, что и прежде, то React предполагает, что DOM-элемент отображает тот же компонент, что и прежде. Но это уже не так.
- **UUID (`universally unique identifier`)** — это стандарт идентификации, используемый в создании программного обеспечения, стандартизированный Open Software Foundation. UUID представляет собой 16-байтный (128-битный) номер. В шестнадцатеричной системе счисления UUID выглядит как: «d29e9b42-9b97-ab6f-bc0c-2c3f938d10e4».

Если у вас нет возможности взять уникальный id со стороны бэкенда вы можете использовать несколько подходов:

- о Использовать библиотеку для генерации uid (к примеру «`shortid`»).
 - о Использовать `singleton` для генерации id (зависит от ситуации).
 - о Использовать `Date.now()` (зависит от ситуации).
 - о Использовать `Math.random()` + `String.replace()`(зависит от ситуации).
 - о Использовать `Crypto API` + `String.replace()` + Арифметический Сдвиг.
- **Разбиение на компоненты** - прежде чем начинать создавать комплексный компонент нужно разбить его на более мелкие части, а также:
 - о Понять какие компоненты нуждаются в `state` а каким будет достаточно `props`.
 - о Вынести состояние на уровень выше если в нем нуждается несколько компонент.

Закрепление материала

- Как указывать тип компонента во время исполнения?
- Зачем нужен `unique id` для атрибута «`key`»?
- Как правильно отрендерить список в React, что для этого нужно?

Дополнительное задание

Задание

Посмотрите пример и попробуйте понять проблему использования [index in key attribute](#).

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Прочитайте документацию react, а именно - раздел с JSX в деталях.

Задание 3

Создайте свой компонент «TODO list», он должен иметь следующее:

- о поле ввода,
- о кнопку для добавления пункта. (Кнопка для добавления пункта должна содержать в себе компонент «Icon», который принимает тип нужной вам иконки в виде строки. (иконка формата .svg))
- о кнопку возле каждого пункта для его удаления.
- о кнопку возле каждого пункта для пометки - сделано. (Должна появиться иконка - галочка слева от пункта.)

Рекомендуемые ресурсы

[Index in key - это анти-паттерн](#)

[Генерация символьных UUID](#)

[JSX в деталях](#)