

38. Компоненты в React



Цель:

Познакомиться со следующими понятиями и возможностями react:

- state в компонентах
- stateless и stateful компоненты
- рендеринг списков
- элементы форм
- условный рендеринг
- подъем состояния

План занятия:

- Понятие state в компонентах
- Что такое stateless и stateful компоненты и в чем их разница
- Списки в react
 - Ключи
 - map() в JSX
- Элементы форм
 - input
 - textarea
 - select
- Условный рендеринг
 - Переменные-элементы
 - Встроенные условия if с логическим оператором &&
 - Встроенные условия if-else с тернарным оператором
 - Предотвращение рендеринга компонента
- Подъем состояния

Конспект:

State в компонентах react:

State это JavaScript объект, который хранит динамические внутренние данные компонента.

- state доступен только внутри компонента
- значения в state можно изменять
- Единственное место, где можно установить объект state - это конструктор класса(если это классовые компоненты) или использовать хук `useState`

Разница между state и props:

- state и props содержат информацию, которая влияет на то, что увидим после рендера, есть существенное различие: props передаётся в компонент (служат как параметры функции), в то время как state находится внутри компонента

Stateless и stateful компоненты:

Компоненты в React содержащие внутреннее состояние — state, называются **Stateful**. Без него — **Stateless**.

Списки в react:

- можно создать коллекцию элементов и встроить её в JSX с помощью фигурных скобок `{}`.
- встроить `map()` в JSX

Чтобы определить в react какие элементы были изменены, добавлены или удалены, мы должны использовать **ключи(keys)**. Их необходимо указывать, чтобы React мог сопоставлять элементы массива с течением времени:

- Ключи нужно определять непосредственно внутри массивов
- Ключи должны быть уникальными среди соседних элементов
- Ключи служат подсказками для React, но они никогда не передаются в ваши компоненты

Элементы форм:

В React HTML-элементы формы ведут себя немного иначе по сравнению с DOM-элементами, так как у элементов формы изначально есть внутреннее состояние.

В HTML элементы формы, такие как `<input>`, `<textarea>` и `<select>`, обычно сами управляют своим состоянием и обновляют его когда пользователь вводит данные.

- `<input type="text">`, `<textarea>`, и `<select>` работают очень похоже. Все они принимают атрибут `value`, который можно использовать, чтобы реализовать управляемый компонент.
- `select` - В HTML `<select>` создаёт выпадающий список.

Условный рендеринг:

React позволяет разделить логику на независимые компоненты. Эти компоненты можно показывать или прятать в зависимости от текущего состояния.

Условный рендеринг в React работает так же, как условные выражения работают в JavaScript.

- **Переменные-элементы:** объявить переменную и сохранить туда элемент и условно рендерить компонент `if`-выражением
- **Встроенные условия `if` с логическим оператором `&&`:**
можно использовать логический оператор `&&` заключив его в фигурные скобки
- **Встроенные условия `if-else` с тернарным оператором:**
можно воспользоваться тернарным оператором `condition ? true : false`, также заключив его в фигурные скобки

Предотвращение рендеринга компонента: В редких случаях может потребоваться позволить компоненту спрятать себя. Чтобы этого добиться, необходимо вернуть `null`.

Подъем состояния:

Очень часто несколько компонентов должны отражать одни и те же данные, которые меняются с течением времени. В таких случаях следует поднимать состояние выше по иерархии: к их ближайшему общему предку.

- Для любых изменяемых данных в React-приложении должен быть один «источник истины».
- Вместо того, чтобы пытаться синхронизировать состояние между различными компонентами, мы должны полагаться на однонаправленный поток данных.
- Такой подход сокращает количество багов и упрощает процесс их поиска, так как один компонент владеет данными и только в нем можно их изменить