

## 41. Хуки в React

### Цель:

Познакомиться со следующими понятиями и возможностями react:

- что такое context и refs
- хуки useContext, useRefs

### План занятия:

- Что такое context
  - использование хука useContext
- Что такое refs
  - использование хука useRefs

## Конспект:

### Context:

Контекст позволяет передавать данные через дерево компонентов без необходимости передавать пропсы на промежуточных уровнях.

Использование контекста: Контекст разработан для совместного использования данных, к которым есть доступ у каждого React-компонента на протяжении всего дерева. По возможности не следует использовать контекст, так как это усложняет повторное использование компонентов.

### Когда использовать контекст:

- ui - тема
- выбранный язык приложения

### React.createContext

**React.createContext** создает объект **context**. Когда React рендерит компонент, который подписан на этот объект, React получит текущее значение контекста из ближайшего подходящего Provider выше в дереве компонентов.

### Context.Provider

**React.Provider.** Каждый объект Context используется вместе с Provider компонентом, который позволяет дочерним компонентам, использующим этот контекст, подписаться на его изменения.

## Хук useContext

Принимает объект контекста (значение, возвращённое из `React.createContext`) и возвращает текущее значение контекста для этого контекста. Текущее значение контекста определяется пропом `value` ближайшего `<MyContext.Provider>` над вызывающим компонентом в дереве.

## Refs

Рефы дают возможность получить доступ к DOM-узлам или React-элементам, созданным в рендер-методе.

### Когда использовать рефы:

Ситуации, в которых использование рефов является оправданным:

- Управление фокусом, выделение текста или воспроизведение медиа.
- Императивный вызов анимаций.
- Интеграция со сторонними DOM-библиотеками.

Важно помнить о рефах:

- Избегать использование рефов в ситуациях, когда задачу можно решить декларативным способом. Например, вместо того чтобы определять методы `open()` и `close()` в компоненте `Dialog`, лучше передавать ему проп `isOpen`.
- Не злоупотреблять рефами

### Доступ к refs:

Когда реф передается элементу в методе `render`, ссылка на данный узел доступна через свойство рефа `current`.

## Хук `useRef`:

**`useRef`** возвращает изменяемый ref-объект, свойство `.current` которого инициализируется переданным аргументом. Возвращенный объект будет сохраняться в течение всего времени жизни компонента.

### Особенности `useRef`:

- `useRef()` может использоваться не только для DOM-рефов. Реф — это общий контейнер, а его свойство `current` — изменяемое и может хранить любое значение, подобно свойству экземпляра класса.
- `useRef` *не* уведомляет вас, когда изменяется его содержимое. Мутирование свойства `.current` не вызывает повторный рендер