

<TeachMeSkills />

# 41. Хуки в React.

## Часть 2

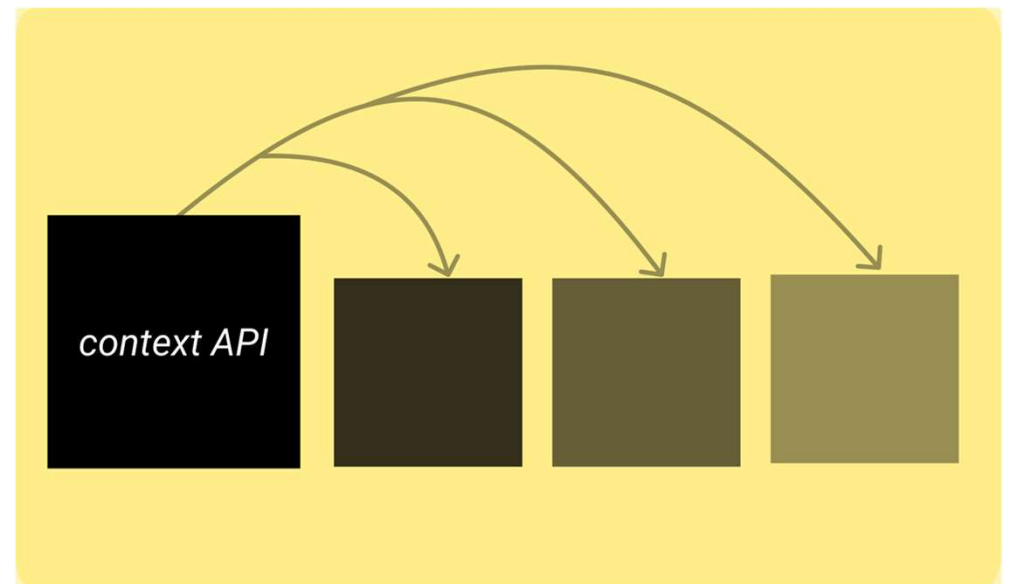
## Цель:

Познакомиться со следующими понятиями и возможностями react:

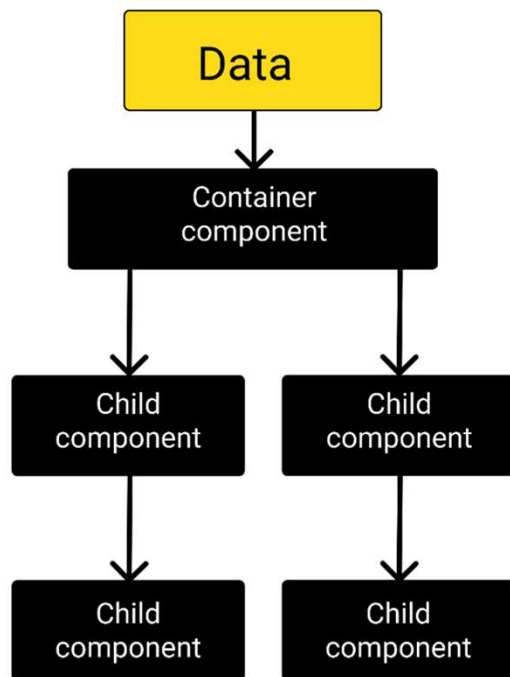
- что такое context и refs
- хуки useContext, useRefs

## Context:

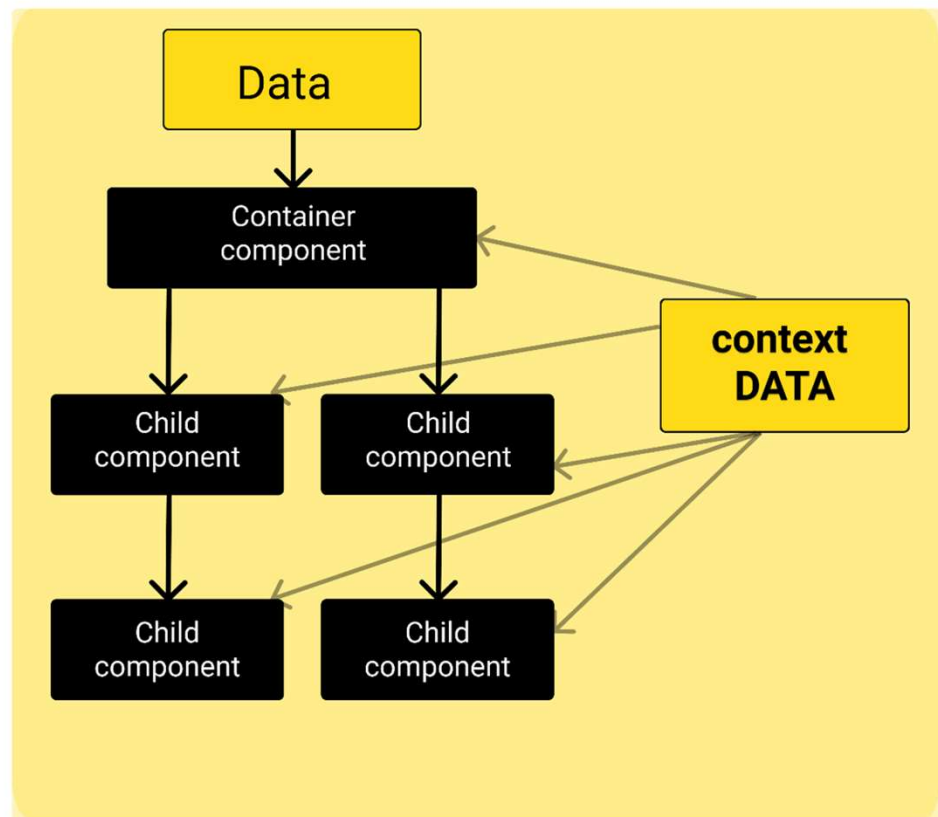
Контекст позволяет передавать данные через дерево компонентов без необходимости передавать пропсы на промежуточных уровнях.



# Context:



*prop drilling*



## 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.

### Примеры использования контекста:

- `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.

```
const MyContext = React.createContext(defaultValue);
```

## Context.Provider:

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

```
<MyContext.Provider value={/* некоторое значение */}>
```

## Хук useContext:

Принимает объект контекста (значение, возвращённое из `React.createContext`) и возвращает текущее значение контекста для этого контекста.

```
const { theme, toggleTheme } = useContext(ThemeContext);

return (
  <div>
    <p>My theme!</p>
    <button onClick={toggleTheme}>
      Switch to {theme === 'light' ? 'dark' : 'light'} mode
    </button>
  </div>
);
```



## Refs:

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

**React.createRef();**

Доступ к рефу: **const node = this.myRef.current;**

## 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.

## Хук 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

```
const inputRef = useRef(initialValue);
```