

# Лабораторная работа: использование `useContext`

В этой лабораторной мы настроим изменение оформления компонента `Basket`

## Упражнение 1: использование контекста для темы оформления

1.1. В файле `src\components\css\Basket.css` добавьте оформление для компонента `Basket`:

```
.Basket-dark h2 {  
  color: #fff;  
}  
  
.Basket-dark * {  
  background: rgb(49, 49, 49) !important;  
  color: #fff;  
}  
  
.Basket-debug * {  
  border: 1px solid #f00 !important;  
}
```

1.2. Создайте файл `src\contexts\theme-context.js` из которого будет экспортироваться вновь созданный контекст

```
import React from "react";  
  
export const themes = {  
  light: "light" ,  
  dark: "dark" ,  
  debug: "debug" ,  
};  
  
export const ThemeContext = React.createContext({  
  theme: themes.debug,  
});
```

1.3. В файле `src\App.js` импортируйте контекст

```
import {themes, ThemeContext} from './contexts/theme-context'
```

а в теле `App` вызовите хук

```
const [theme, setTheme] = useState(themes.light);
```

#### 1.4. Оберните `{basketPlace}` в `<ThemeContext.Provider>`

```
<ThemeContext.Provider value={{theme, setTheme}}>
  {basketPlace}
</ThemeContext.Provider>
```

#### 1.5. Внесите изменения в `src\components\BasketItem.js` - нужно импортировать `useContext`, импортировать сам контекстный-объект `import { ThemeContext } from "../contexts/theme-context"` и потом использовать основное свойство контекста `theme`

```
import React, { useContext } from "react";
import "../css/BasketItem.css";

import ItemInfo from "../ItemInfo";
import ItemPrice from "../ItemPrice";
import Counter from "../Counter";

import { ThemeContext } from "../contexts/theme-context";

const BasketItem = ({
  uid,
  title,
  description,
  price,
  qty,
  setItems,
  items,
}) => {
  const myContext = useContext(ThemeContext);
  const { theme } = myContext;

  return (
    <div className={"BasketItem BasketItem-" + theme}>
      <ItemInfo title={title} description={description} />
      <ItemPrice value={price} currency={"P"} />
      <Counter value={qty} uid={uid} items={items} setItems={setItems} />
      <ItemPrice value={qty * price} currency={"P"} />
    </div>
  );
};

export default BasketItem;
```

#### 1.6. Обновите страницу и посмотрите на код в отладчике браузера - у элементов корзины будет виден класс `BasketItem-light`

```

<div class="BasketItem" style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">
  <div class="BasketItem BasketItem-light" style="background-color: #f9f9f9; padding: 10px; border: 1px solid #eee; border-radius: 5px; flex-direction: column; align-items: center;">
    <div class="ItemInfo">
      <h3 class="ItemInfo_title">Товар1</h3>
      <p class="ItemInfo_description">описание товара 1</p>
    </div>
  </div>
</div>

```

### 1.7. Импортируйте контекст в `src\components\Basket.js`

```
import { themes, ThemeContext } from "../contexts/theme-context";
```

### 1.8. Добавьте в компонент `Basket` кнопки переключения темы

```

...
return (
  <ThemeContext.Consumer>
    ({{theme, setTheme}} => {

      return <div className={"Basket " + 'Basket-' + theme}>

        ...

        <Button
          value="Продолжить покупку"
          onClickHandler={() => alert("Продолжить")}
          className="btn-proceed"
        />
        <Button
          value="Тёмная тема"
          onClickHandler={() => setTheme(themes.dark)}
        />
        <Button
          value="Светлая тема"
          onClickHandler={() => setTheme(themes.light)}
        />
        <Button
          value="Отладка"
          onClickHandler={() => setTheme(themes.debug)}
        />
      </div>
    ) }
  </ThemeContext.Consumer>
);

```

### 1.9. Переключитесь на тёмную тему и убедитесь, что она выглядит как показано на изображении

Ваша корзина

4

<div>Товар1</div> <div>описание товара 1</div>	1200₽	<div>-</div> <div>1</div> <div>+</div>	1200₽
<div>Товар2</div> <div>описание товара 2</div>	800₽	<div>-</div> <div>1</div> <div>+</div>	800₽
<div>Товар3</div> <div>описание товара 3</div>	250₽	<div>-</div> <div>2</div> <div>+</div>	500₽
<div>Промокод</div> <div>REACTSPECIALIST</div>	-500₽		
<div>Всего</div> <div>(РУБ)</div>	2000₽		

Промокод

Применить

Продолжить покупку

Темная тема

Светлая тема

Отладка