1. Какую ошибку я допустила в этом примере?

Ответ: backgroundColor: 'red'

2. Какие есть способы работы со стилями в React?

Ответ: Метод 1: Инлайновая стилизация
Метод 2: Таблицы стилей CSS
Метод 3: CSS модули
Метод 4: CSS-препроцессоры

3. Как будет выглядеть карточка товара, если ей передать аттрибут `addedToCart===0`?

Ответ:Если аттрибут не передан, значит товар в корзину не добавлен.

4. Какие еще проверки нужно было бы сделать для аттрибута `addedToCart`?

Ответ: что перед нами число, а не имя котика

5. Клиент попросил повесить тег "New" на товары из новой коллекции. Как это сделать, какой условный оператор выбрать?

Ответ:Логический оператор &&

6. Какими тремя способами можно написать условный рендеринг?

Ответ:1.If 2.Логический оператор && 3.Тернарный оператор

7. Представьте, что вы пишете компонент логина. Если пользователь авторизован, то мы показываем его имя, а если нет, то даем возможность ввода логина и пароля. Какой код для этого нужно написать, если за авторизацию пользователя отвечает флаг isAutorized?

Ответ:
```
function Login({ isAuthorized }) {
  // Создаем состояние для хранения логина и пароля
  const [login, setLogin] = useState('');
  const [password, setPassword] = useState('');

  // Обработчик изменения поля логина
  const handleLoginChange = (event) => {
    setLogin(event.target.value);
  };

  // Обработчик изменения поля пароля
  const handlePasswordChange = (event) => {
    setPassword(event.target.value);
  };

  // Если пользователь авторизован, показываем его имя
  if (isAuthorized) {
    return (
      <div>
        Добро пожаловать, Имя Пользователя!
      </div>
    );
  }

  // Если пользователь не авторизован, показываем форму ввода логина и пароля
  return (
    <form>
      <div>
        <label htmlFor="login">Логин:</label>
        <input type="text" id="login" value={login} onChange={handleLoginChange} />
```

```
      </div>
      <div>
        <label htmlFor="password">Пароль:</label>
        <input type="password" id="password" value={password}
onChange={handlePasswordChange} />
      </div>
      <button type="submit">Войти</button>
    </form>
  );
}
```

Компонент Login получает пропс isAuthorized, который определяет, авторизован ли пользователь или нет. Если пользователь авторизован, компонент возвращает элемент с сообщением "Добро пожаловать, Имя Пользователя!". Если пользователь не авторизован, компонент возвращает форму для ввода логина и пароля. В форме используются обработчики изменения состояния, чтобы обновлять значения логина и пароля при их изменении пользователем.

8. В чем преимущества использования препроцессоров? Какой есть еще способ использовать переменные, кроме $ в препроцессорах?

Ответ: **@**

Также в SASS существуют операторы, позволяющие выполнять арифметические и логические операции с переменными. Например, можно складывать, вычитать, умножать и делить числовые переменные, а также использовать логические операторы, такие как and, or и not.
Наконец, в SASS существует возможность использовать переменные внутри селекторов и вложенных блоков кода. Это позволяет писать более компактный и удобочитаемы код, а также легко изменять стили для различных элементов страницы, используя одни и те же переменные.

- Разделение на небольшие файлы. Мы можем организовать блочную систему стилей
  - Переменные. Вынести общие ****(цвета, размеры шрифта, и т. д.) и импортировать их в нужный файл
  - Математические операции
- **Вложения.** Позволяет не повторять написание селектора, внутри которого нужно что-то стилизовать. Просто пишем вложенные селекторы внутри родительских. Ощутимо ускоряет работу.
- **Примеси.** Набор правил, который можно использовать многократно.
- **Амперсанд.** Позволяет не повторять родительский селектор, быстро менять его. Помогает сделать код читабельней.