### Вопросы 💎

1. Какую ошибку я допустила в [этом](https://www.notion.so/27-076f6e0f9f454440be8ec3c04b61789c) примере?

//

1. Какие есть способы работы со стилями в React?
   * Инлайновая стилизация, то есть когда стили прописываются как атрибут style={{…}} внутри JSX элемента, или прописываются в переменную, которая затем передается как объект атрибута style.
   * Таблицы стилей CSS (прописываем стили к className, присваиваем className к нужным элементами, и импортируем таблицу в компонент и применяем стили стандартным образом).
   * Через CSS модули. Помогают решить проблему с совпадением имен. Работают по принципу компиляции. К названию файла стилей добавляется слово module, например: input.module.css. В остальном все как в таблице стилей, просто на выходе мы получим уникальное название класса, которое не задублируется с другим.
   * Через темы (темизация). Стиль становится свойством компонента, мы не импортируем его (как таблицу стилей) в компонент.
   * CSS – переменные, благодаря которым можно уменьшить повторения, улучшить читаемость и добавить гибкость стилям (похоже на sass, однако можно не подключать препроцессор, и создав одну переменную с определенным стилем, применить ее в нескольких местах.
   * Через CSS-препроцессоры (SASS, LESS).
2. Как будет выглядеть карточка товара, если ей передать аттрибут addedToCart===0?

//

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

//

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

//

1. Какими тремя способами можно написать условный рендеринг?

//

1. 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?

//

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

* Использование переменных для общих стилей.
* Возможность прямо в стилях прописывать математические операции (calc).
* Вложенность (можно прописывать стили внутри родительского селектора, по тегам), не прописывая отдельно классы селекторов.
* Использование @mixin (правила для неоднократного использования), например, одинаковые псевдоклассы или псевдоэлементы для кнопок на сайте и др.
* Амперсанд & (использовать вместе с вложением, вместо родительского селектора указать &).