Код  
**e.preventDefault()** - это метод, используемый в JavaScript для предотвращения стандартного поведения события в браузере.

Когда вы отправляете форму веб-страницы, браузер по умолчанию обновляет страницу, чтобы отправить данные формы на сервер и отобразить ответ. Однако в некоторых случаях, например, при использовании AJAX-запросов для отправки данных на сервер без перезагрузки страницы, вы можете захотеть отменить это стандартное поведение, чтобы иметь больше контроля над тем, что происходит после отправки формы.

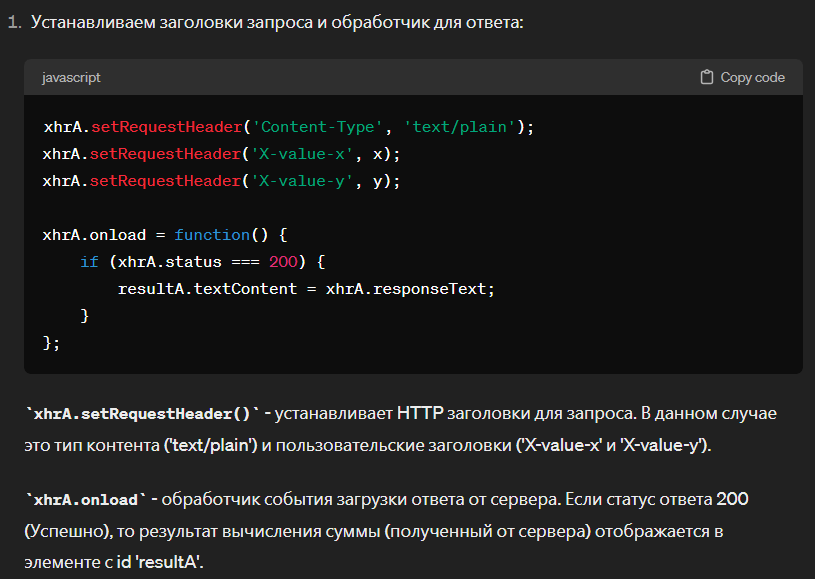
**const res = await fetch('/sum', { ... });**: Этот код отправляет запрос на сервер и ждет его завершения. Когда запрос выполнится, результат будет сохранен в переменной **res**.

**res.headers.get('X-Value-z')**: Это метод используется для извлечения значения конкретного заголовка из ответа сервера.

const count = Math.floor(Math.random() \* (10 - 5 + 1)) + 5;

**Math.random()** возвращает случайное число от 0 до 1 (не включая 1).

**Math.random() \* (верхняя\_граница - нижняя\_граница + 1)** даст случайное число в диапазоне от 0 до **(верхняя\_граница - нижняя\_граница + 1)**.  
**Math.floor(Math.random() \* (верхняя\_граница - нижняя\_граница + 1))** округляет это число вниз до ближайшего целого числа.



Теория

технология AJAX (Asynchronous JavaScript and XML), которая основывалась на возможности доступного из JavaScript объекта XMLHTTPRequest асинхронно отправлять http-запросы серверу и обрабатывать его ответы

Вот основные методы и свойства объекта **XMLHttpRequest**:

1. **new XMLHttpRequest()**: Создает новый экземпляр объекта **XMLHttpRequest**.
2. **XMLHttpRequest.open(method, url, async)**: Инициализирует запрос. Этот метод принимает три аргумента: метод запроса (GET, POST и т. д.), URL-адрес, по которому отправляется запрос, и флаг асинхронности (по умолчанию **true**, что означает асинхронный запрос).
3. **XMLHttpRequest.setRequestHeader(header, value)**: Устанавливает значение HTTP заголовка для запроса.
4. **XMLHttpRequest.send(data)**: Отправляет запрос на сервер. В случае POST-запроса, данные могут быть переданы в качестве аргумента этого метода.
5. **XMLHttpRequest.onreadystatechange**: Свойство-обработчик, которое вызывается каждый раз, когда изменяется состояние объекта **XMLHttpRequest**. Обычно используется для отслеживания изменений состояния запроса.
6. **XMLHttpRequest.readyState**: Свойство, содержащее текущее состояние объекта **XMLHttpRequest**. Возможные значения:
   * **0**: **UNSENT** - объект создан, но метод **open()** еще не вызван.
   * **1**: **OPENED** - метод **open()** был вызван.
   * **2**: **HEADERS\_RECEIVED** - метод **send()** был вызван и заголовки ответа получены.
   * **3**: **LOADING** - загрузка данных; **responseText** содержит частичные данные.
   * **4**: **DONE** - операция завершена; все данные получены.
7. **XMLHttpRequest.responseText**: Свойство, содержащее текстовое содержимое ответа сервера.
8. **XMLHttpRequest.responseXML**: Свойство, содержащее XML-документ ответа сервера. Это полезно, если сервер возвращает XML-данные.

**onload** - это событие в JavaScript, которое срабатывает, когда загружается вся страница или определенный ресурс