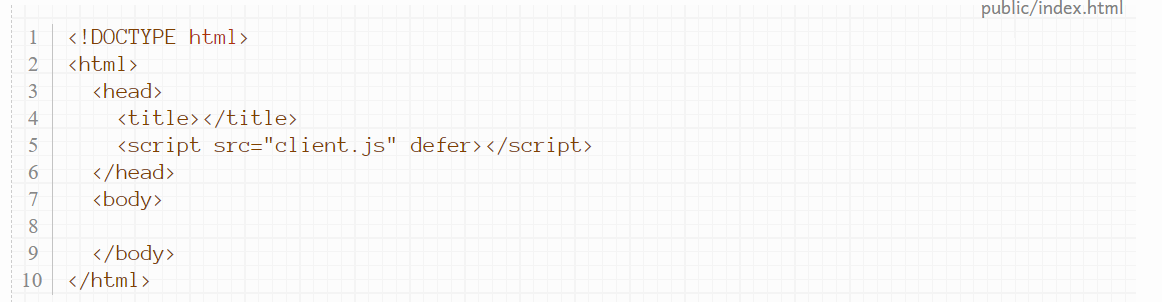
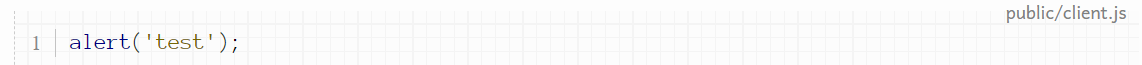
**Введение в AJAX**

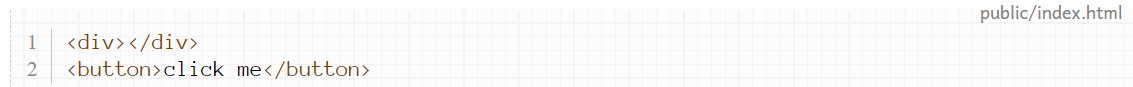
Для изучения работы AJAX необходимо наличие сервера. Сделаем в папке с сервером HTML файл, который мы будем открывать в браузере:



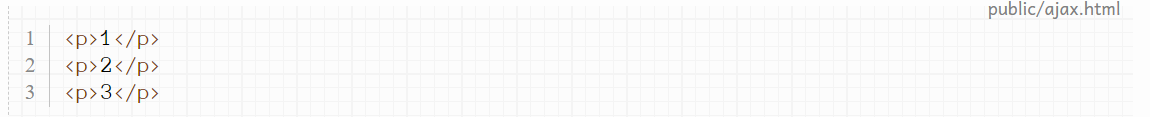
Подключим к нему JavaScript файл:



Давайте теперь посмотрим, как подгрузить часть страницы с помощью AJAX. Пусть на нашей странице index.html будет расположен див и кнопка (здесь показано содержимое body):

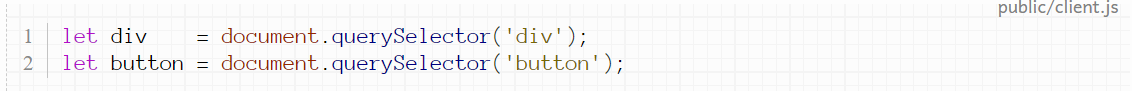


Пусть у нас также есть страница ajax.html, на которой расположены три абзаца (без тегов страницы, просто абзацы):

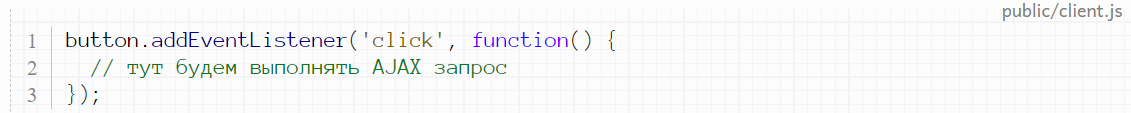


Давайте сделаем так, чтобы по клику кнопку на странице index.html в див подгрузилось содержимое страницы ajax.html.

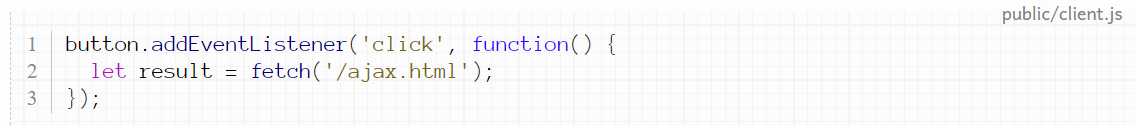
Для этого для начала получим в переменные ссылки на наши элементы:



Давайте навесим на кнопку обработчик клика:

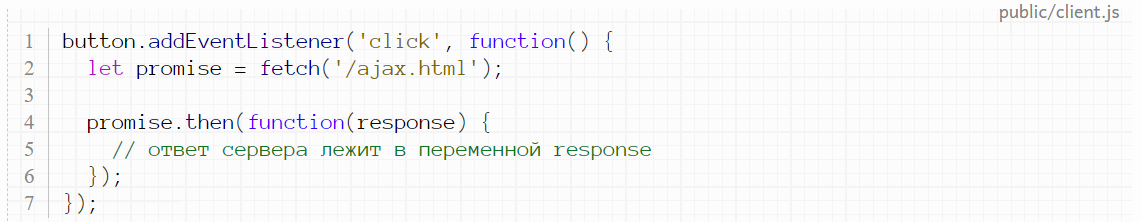


Давайте теперь выполним AJAX запрос. Для этого используется функция fetch, параметром получающая адрес страницы, содержимое которой мы хотим получить. Укажем адрес страницы ajax.html на нашем сервере:

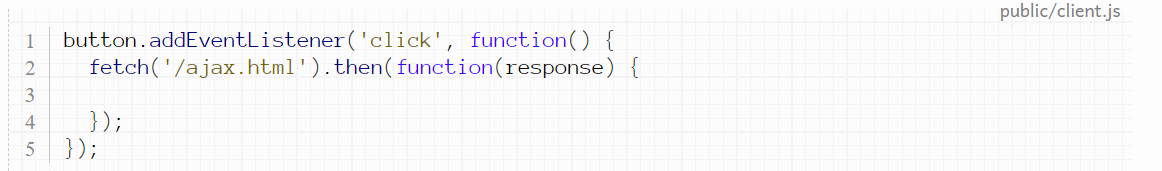


Своим результатом fetch вернет [промис](https://code.mu/ru/javascript/book/supreme/promises/intro/). Дело в том, что запрос страницы через AJAX - это асинхронная операция, ведь пройдет некоторое время, через которое страница нам ответит.

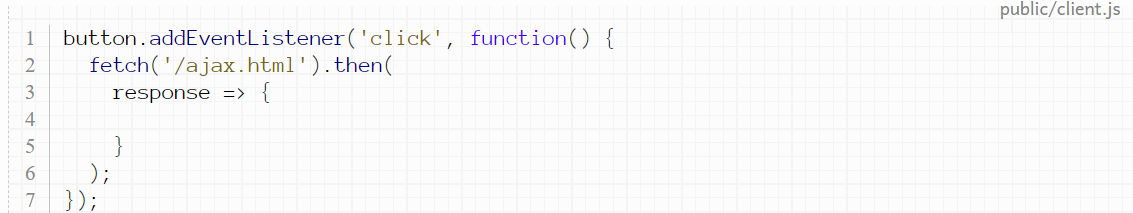
Давайте получим результат промиса:



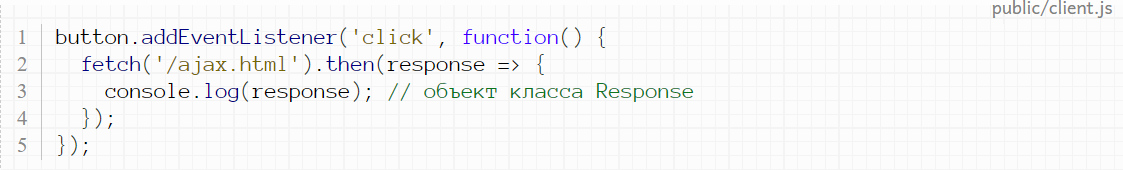
Упростим, избавившись от лишней переменной:



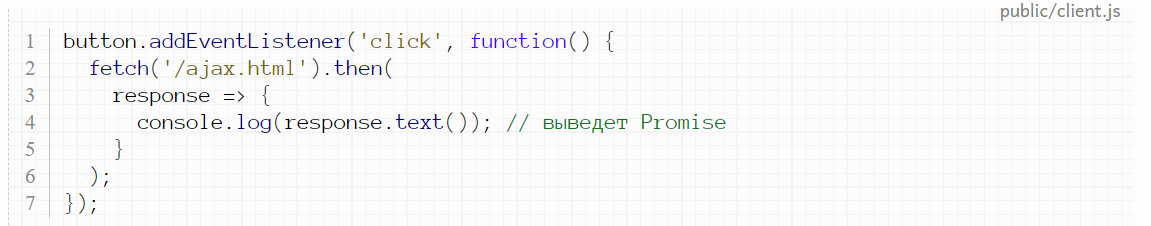
Упростим, воспользовавшись стрелочной функцией:



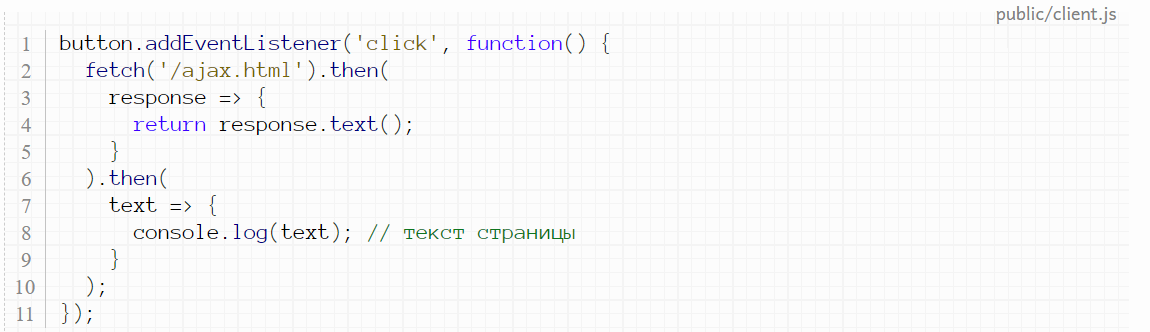
В переменной response содержится достаточно сложный объект класса Response:



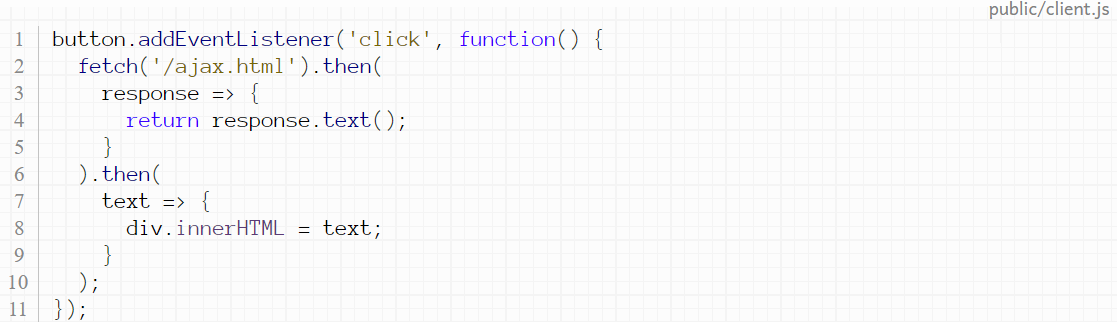
Нам нужно получить текст запрошенной страницы. Для этого в этом объекте существует метод text. Этот метод, однако, возвращает не текст страницы, а промис с ее текстом:



Чтобы добраться до текста страницы, необходимо промис, полученный из response.text, обработать еще раз:



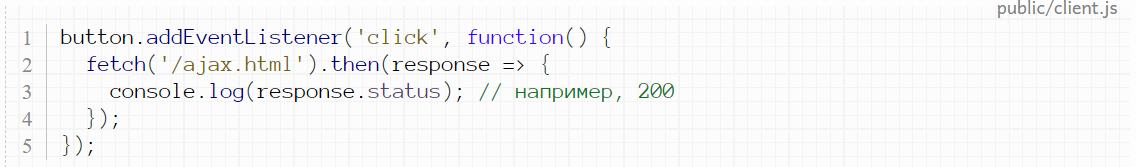
Итак, мы наконец получили текст запрошенной страницы и можем, например, записать его в наш див:



**Выполните самостоятельно:**

Даны три кнопки и див. На сервере даны три страницы. Сделайте так, чтобы каждая из кнопок подгружала в див соответствующую страницу.

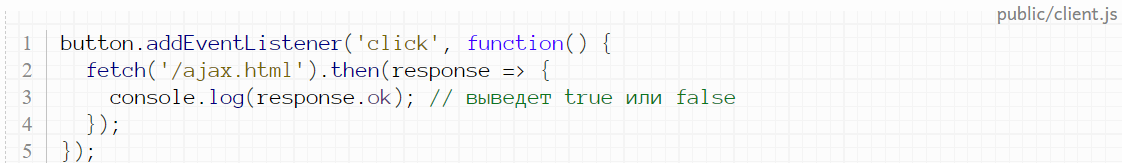
В свойстве status ответа response содержится код HTTP ответа:



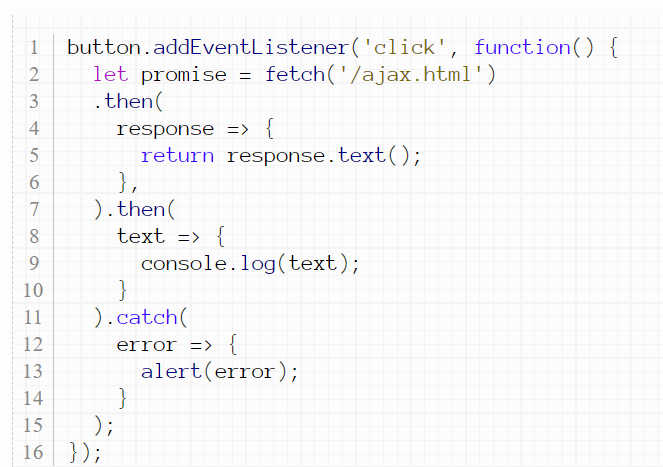
**Выполните самостоятельно:**

Выведите текст страницы, если получен статус 200, и ошибку, если получен статус 404.

В свойстве ok содержится true, если сервер ответил с HTTP статусом из группы 2xx, и false, если из другой группы:

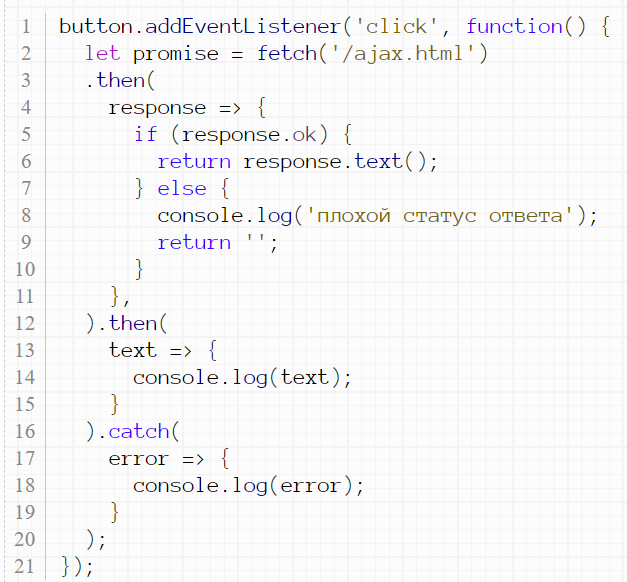


Если при выполнении AJAX запроса случится исключительная ситуация (например, оборвется интернет), то промис завершится с ошибкой. Мы сможем поймать эту ошибку удобным нам способом, например, через catch:

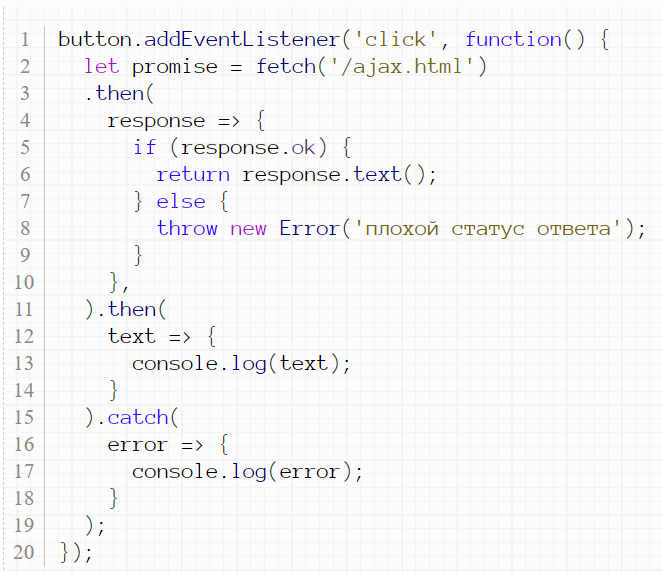


Промис, который возвращает fetch, завершается с ошибкой, только если произошла ошибка сети. Если же сервер вернул ответ со статусом 404 или 500, то промис будет завершен успешно, но при этом статус ok будет установлен в false.

Давайте перехватим оба типа ошибок:



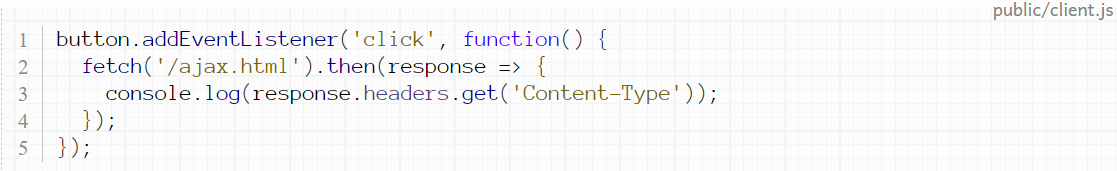
Сделаем так, чтобы ошибка, связанная с плохим статусом HTTP ответа, также ловилась блоком catch. Для этого прокинем ее дальше через throw:



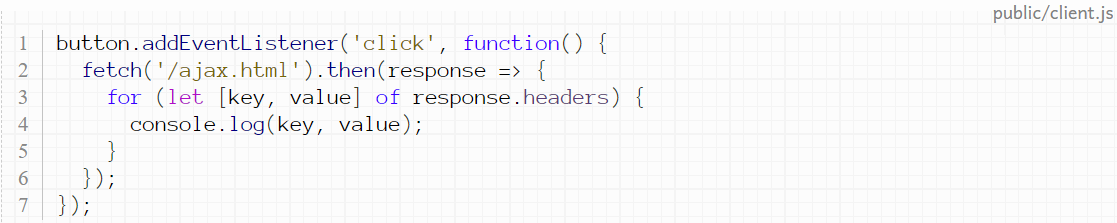
**Выполните самостоятельно:**

Выведите текст страницы, если запрос был успешным, и ошибку, если что-то пошло не так.

Заголовки ответа хранятся в свойстве headers объекта response. Давайте получим какой-нибудь заголовок:



А теперь переберем все заголовки циклом:



**Выполните самостоятельно:**

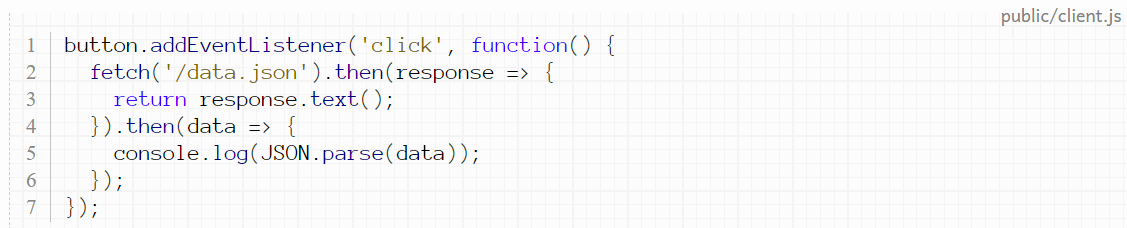
Получите содержимое заголовков Content-Language и  Content-Length.

**Получение JSON через AJAX в JavaScript**

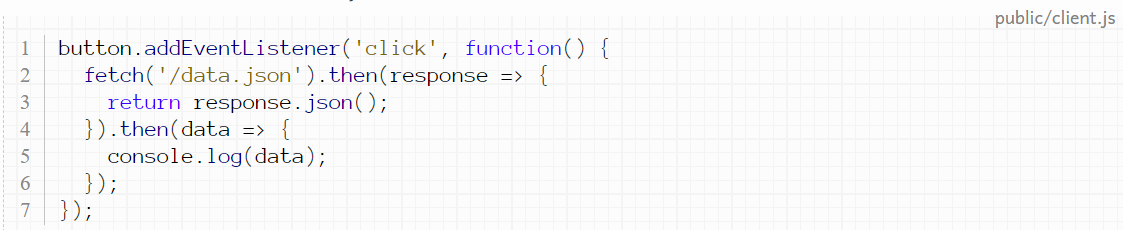
Распространенной практикой является отдача сервером данных в формате JSON. Давайте посмотрим, как с этим работать. Пусть на сервере есть следующий файл:



Давайте получим данные из этого файла:



Можно воспользоваться методом json, чтобы JSON автоматически преобразовался в структуру JavaScript:



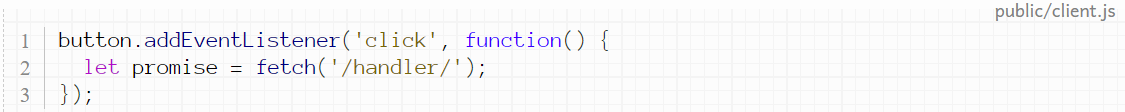
**Выполните самостоятельно:**

Пусть сервер отдает JSON с массивом. Переберите этот массив циклом и запишите каждый его элемент в отдельную li.

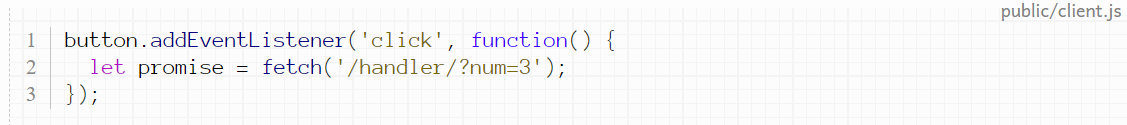
# Отправка GET запросов через AJAX в JavaScript

Давайте теперь научимся передавать данные на сервер в AJAX запросе. Сервер что-то будет делать с этими данными и отправлять нам обратно результат.

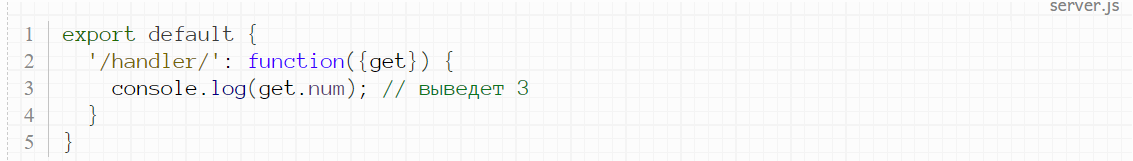
Наш тестовый сервер будет обрабатывать данные, отправленные на адрес /handler/:



Для начала давайте отправим GET запрос:



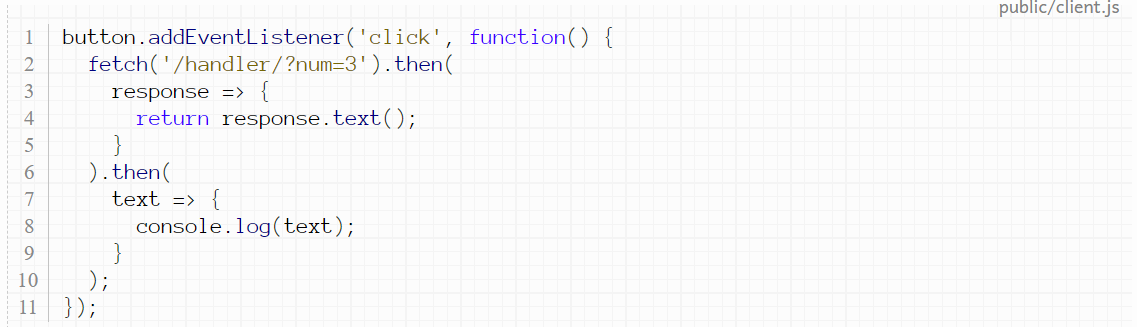
Давайте получим отправленные данные на стороне сервера:



Давайте что-нибудь сделаем с этими данными и отправим их обратно:



На стороне клиента получим результат и выведем его куда-нибудь:

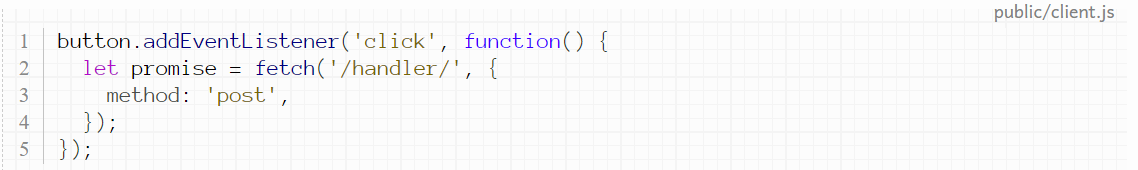


**Выполните самостоятельно:**

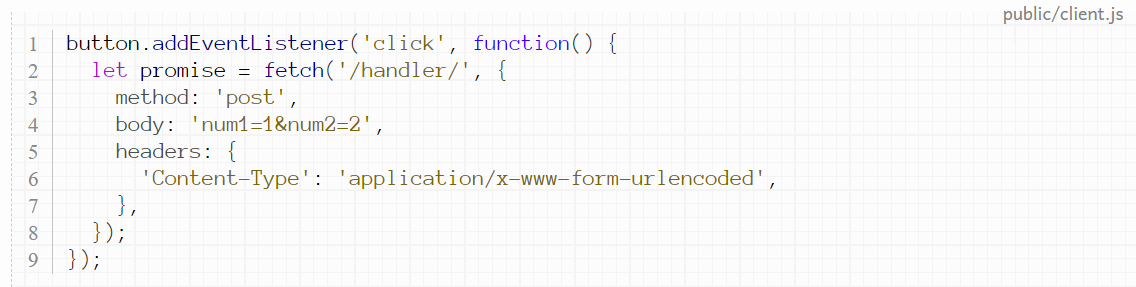
1. На клиенте дан див и кнопка. По нажатию на кнопку отправьте на сервер два числа. Пусть сервер найдет сумму переданных чисел. Результат запишите в див.
2. Пусть на сервере дан массив. Пусть сервер ожидает, что параметром будет передано число, и возвращает элемент массива, соответствующий этому числу. По нажатию на кнопку передайте на сервер некоторое число, а ответ сервера выведите в абзац.

**Отправка POST запросов через AJAX в JavaScript**

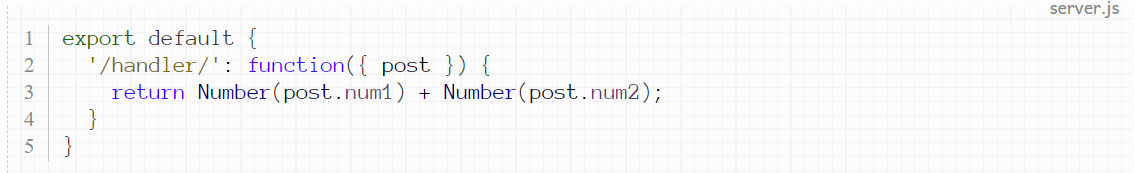
Давайте теперь отправим на сервер POST запрос. Для этого у функции fetch есть второй параметр с настройками. В настройке method указывается метод HTTP запроса. Укажем метод POST:



Давайте также зададим данные, которые мы хотим отправить на сервер. В POST запросах данные передаются в теле HTTP запроса. Для этого данные нужно указать в настройке body. Укажем их в виде Query String, задав их тип соответствующим MIME:



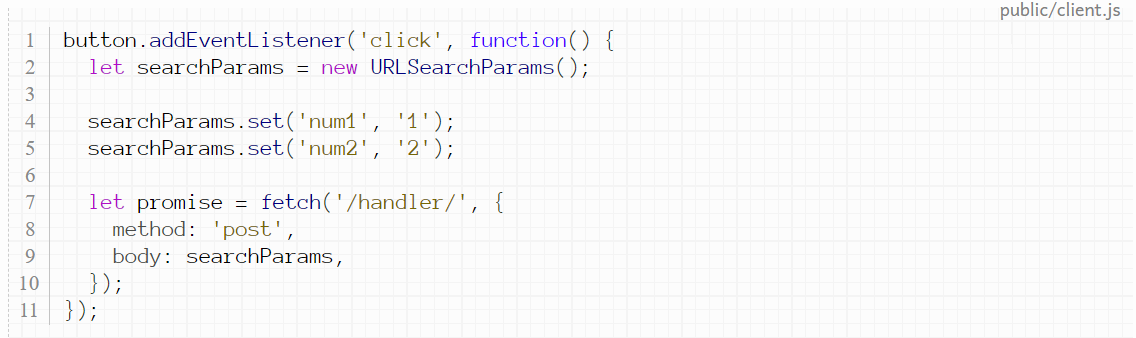
Получим эти данные на сервере, сделаем с ними что-нибудь и отправим их обратно:



**Выполните самостоятельно:**

На клиенте дан див и кнопка. По нажатию на кнопку отправьте на сервер три числа методом POST. Пусть сервер найдет сумму переданных чисел. Результат запишите в див.

Часто не очень удобно формировать вручную Query String. Более удобно это сделать через [URLSearchParams](https://code.mu/ru/javascript/book/supreme/http/URLSearchParams/):



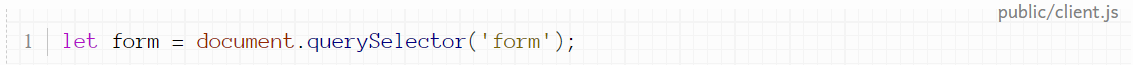
**Отправка формы через AJAX с помощью FormData JavaScript**

С помощью FormData можно автоматически отправить все поля формы на сервер. Давайте посмотрим, как это делается.

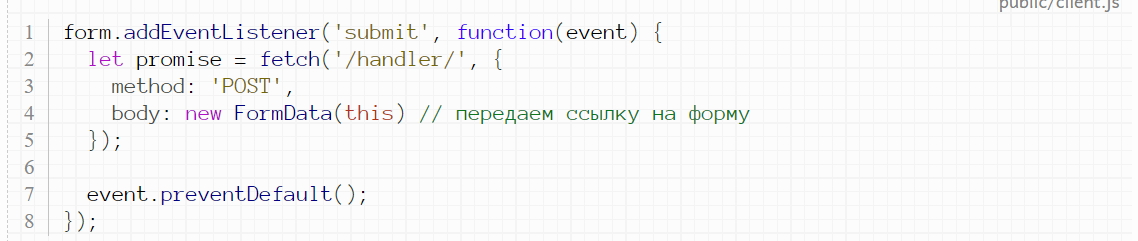
Пусть у нас есть следующая форма, в которой инпутам даны атрибуты name:



Давайте отправим эту форму через AJAX. Для начала получим ссылку на форму в переменную:



Теперь в теле запроса в качестве данных укажем объект FormData. А самому объекту параметром передадим ссылку на форму:



В результате на сервере мы сможем получить значения инпутов по их именам из атрибутов name:

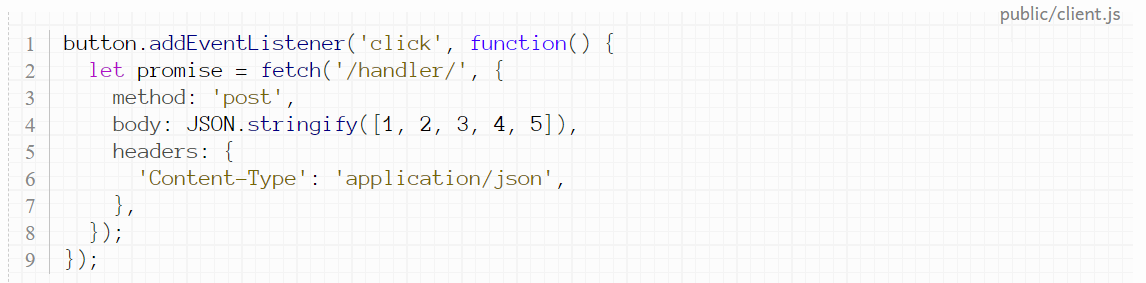


**Выполните самостоятельно:**

Дана форма с пятью инпутами, в которые вводятся числа. Отправьте эту форму на сервер методом POST. Пусть сервер найдет среднее арифметическое введенных чисел и отправит результат обратно в браузер.

**Отправка JSON на сервер с помощью AJAX в JavaScript**

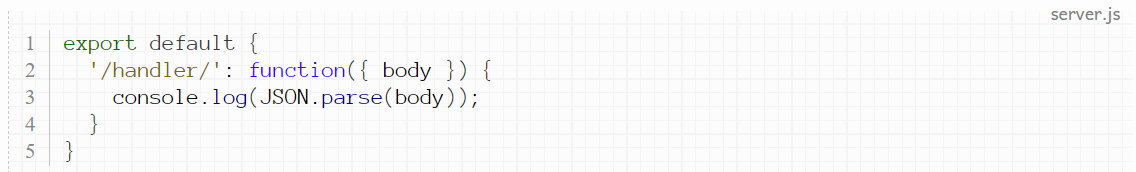
На сервер можно отправлять данные в виде JSON. Это делается следующим образом:



На сервере тело запроса будет лежать в body:



Распакуем полученный сервером JSON:



**Выполните самостоятельно:**

Отправьте на сервер массив с данными. Пусть сервер найдет сумму элементов этого массива и отправит обратно.